

AD A111105



LEVEL

AD A111105

DTIC
ELECTRONIC
FEB 19 1982
S A



DTIC FILE COPY

UNITED STATES AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

This document has been approved
for public release and sale; its
distribution is unlimited.

Copy available to DTIC does not
permit any legible reproduction

82 02 18 121

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

AFIT/GOR/AA/81D-1

MADAM:
MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL
VOLUME II
THESIS

AFIT/GOR/AA/81D-1

Wayne A. Stimpson
2Lt USAFR

DTIC
SELECTE
FEB 19 1982
A

This document has been approved
for public release and sale; its
distribution is unlimited.

Thesis

MADAM:

MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL

Volume II

by

Wayne A. Stimpson
2Lt USAFR

Prepared in partial
fulfillment of the
requirements for a
Masters Degree

December 1981

School of Engineering
Air Force Institute of Technology
Wright-Patterson Air Force Base
Ohio



Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
P	
DISTRIBUTION	
17-12-1981	
A 23	
GP	

Volume II

Table of Contents

List of Figures and Tables	iii
Abstract	iv
I. Introduction.	1
II. Data Structures	2
III. Sensitivity Analysis.	7
IV. Hierarchy Manipulation.	9
Subroutines NODIN and FIND.	9
Subroutines PRETOT, PRENEX, and NEXT.	11
Subroutine SPAN	13
Subroutine CALC	13
V. Variable List	15
Variables Used.	15
Pseudo-variables.	18
Others.	18
Cross Reference Map	19
VI Program Structure	20
Program Source Code.	23

List of Figures and Tables

	<u>Page</u>
Table 1 Pseudo-array Variables	3
Figure 1 Data Fields for Node Record.	4
Table 2 Record Content	4
Figure 2 The Search Process (FIND).	10
Figure 3 Logic Structure for NEXT	12
Figure 4 Logic Flow of CALC	14

Abstract

The complex multifaceted decision situations present today suggest the need for a timely, automated tool. A decision-maker is forced into comparing alternative actions or systems over an entire set of different measures of merit. This effort is an on-line, real-time, computer-based decision aid designed to assist the decision-maker in clarifying preferences in a complex decision environment. It is applicable to problems which may be represented by a hierarchy of objectives to be satisfied. The program is MADAM: Multiple-Attribute Decision Analysis Model, and it is written in FORTRAN V and is implemented on the CYBER 175 system. MADAM is designed to aid the decision-maker as he or she progresses through problem formulation, parameterization, sensitivity analyses, and a decision, including storage of all data and rationales. Deterministic problems are analyzed through Multi-Attribute Utility Theory concepts and an additive value function is utilized for sensitivity analysis. Pairwise preferential independence is tested between attributes. The sensitivity analyses include a cumulative weight analysis, a relative weight analysis, and an attribute level analysis. The analyses may be conducted by fixing an objective to be considered and conducting the analysis across the alternative systems or actions, or conversely by fixing the alternative to be considered and conducting the analysis across a desired set of objectives.

The work is divided into two volumes. Volume I is a theoretical presentation and includes a user's manual. It requires no programming expertise and may be used independently of Volume II. Volume II is a programming manual including the source code. It may not be used independently of Volume I.

I INTRODUCTION

This manual is designed to supplement the information contained in Volume I of this thesis. The fundamental intention to this volume is to provide decision analysts with an elucidation of the details of MADAM in order to better understand the program and, hence, facilitate its implementation. Future analysts may also have the desire to modify or extend the scope of problems suitable for analysis with MADAM, and the following information will be very useful.

MADAM is written in FORTRAN V and implemented on the CYBER 175 system. An effort was made to keep the code as machine independent as possible, and for those situations where this was not possible, an explicit discussion concerning modifications pre-requisite to transporting MADAM IS provided. All references to an example problem will refer to the example illustrated in Chapter V of Volume I.

II DATA STRUCTURES

MADAM utilizes a random access file for storing and manipulating data. Throughout the program are pairs of subroutines and functions which are responsible for setting and retrieving data. This is possible because FORTRAN V will not recognize an array variable until it is determined that no function by that name exists. Thus, for example, MADAM uses the function ARAY (I, J) to represent a pseudo-array variable. A complete list of the subroutine function pairs which represent psuedo-array variables is given in Table 1. Each record of the random access file is 100 words in length. This 100 word record is divided up in to data fields as shown in Figure 1 for the first 500 records. The remaining records have data fields as shown in Table 2.

NOTE***The current ability to access both real and integer (character) data from a record is dependent on the fact that the CYBER 175 system uses one word for both types of information. In order to implement MADAM on a system which uses two words for real data and one word for integer data, it would be necessary to modify the psuedo-array variables so that the desired words will be accessed. This will require manipulation of the record length and storage (retrieval) subscripts.

Each data file (tree) constructed will require about 300 PRU for permanent storage.

As illustrated in Figure 1, each node record contains information about the position of the node in the objective hierarchy plus any node-

Pseudo-array Variable	Associated Subroutine	Associated Function	Information Generated by Variable Relative and Cumulative Weight of Node Best/Worst Levels of Attributes Attribute Names Comments or Rationale Node Position Information Attribute Name at Node 1 Objective at Node 1 Regression Parameters (Value Functions) Alternative System Names System Values at Node 1
ARRAY (I,J)	ASET (I,J,V)	ARRAY (I,J)	
ATT (I,J)	ATTVAL (I,J,V)	ATT (I,J)	
ATT1 (I)	ATTLBL (I,ATTRIB)	ATT1 (I)	
CRAY (I)	CSET (I)	CRAY (I) (Subroutine)	
IRAY (I,J)	ISET (I,J,IM)	IRAY (I,J)	
LABEL (I)	LSET (I,LABELT)	LABEL (I)	
OBJECT (I,OBJCTV)	OSET (I,OBJCTV)	OBJECT (I,OBJCTV) (Subroutine)	
PARAM (I,J)	PSET (I,J,V)	PARAM (I,J)	
SYSLBL (I)	SYSSET (I,SYSLBL)	SYSLBL (I)	
VRAY (I,J)	VSET (I,J,V)	VRAY (I,J)	

Table 1 Pseudo-array Variables

Data	Word(s)	Variable(s)
Node Reference Number Digit	1	IRAY (I,1)
First Descendant	2	IRAY (I,2)
First Crosslink	3	IRAY (I,3)
Backlink	4	IRAY (I,4)
Relative Weight	5	ARAY (I,1)
Cumulative Weight	6	ARAY (I,2)
Associated Attribute	7	LABEL (I)
System Values	8 to (NSYS+7)	VRAY (I,J)
Comments (Rationale)	(NSYS+7) to 64	-none-
Objective	65 to 80	OBJECTV (1) to OBJECTV (16)

Figure 1 Data Fields for Node Record

Information	Record No.
Node Record	1 to 500
Number of Nodes (NNODES)	501
Number of Attributes (NATT)	502
Number of Systems (NSYS)	503
Null Record	504
Regression Intercepts	505
Regression Slopes	506
Sum of Squared Errors	507
Regression Form Indicators	508
System Names	509
Attribute Names	510
Null Record	511
Worst Attribute Levels	512
Best Attribute Levels	513

Table 2 Record Content

specific data. The positional information is provided by the first four words in the record. These words contain the node reference number digit, the first descendant, the first crosslink, and the backlink respectively. The node reference number digit indicates where this node is on its span. For instance, in the sample problem, the node reference number digit of the node with the objective: to maximize aircraft survivability, is two. The first descendant pointer contains a number of the first child of the given node (if any). Absence of a first child ($IRAY(I,2) \leq 0$) implies that the given node is a data node. The first crosslink pointer contains a number of the immediate siblings of the given node (if any). The backlink pointer contains the number of the node which has this given node as a first descendant or a first crosslink.

The remaining words in a node record provide the data for identifying the node and incorporating it into the decision analysis. The fifth word of the record ($ARRAY(1,1)$) contains the weight of the given node relative to its siblings. These relative weights are normalized to a unity sum across the span. The sixth word of a node record ($ARRAY(I,2)$) contains the cumulative weight of a given node. It is automatically calculated by MADAM during the "folding-back" of the objective hierarchy (see subroutine CALC), and it is a function of the node's relative weight and the cumulative weight of its parent. The seventh word in a node record ($LABEL(I)$) contains the attribute label to be associated with that node. It is relevant only for data nodes, and is left a null field for all nodes interior to the tree.

NOTE***All character data is designed around the CYBER's 10 letters/word format. In order to implement MADAM on a system with a different

storage format, it will be necessary to scale all input data or data structures to incorporate a different number of characters per word.

The next set of words in a node record (VRAY (I,J)) hold the values of the alternative systems relative to a given node. These values are either calculated by MADAM (for an interior node) or indirectly input by the user (for a data node). Only as many words as needed are used for this information based on the number of systems. Those words between the system values and the 64th word are used to hold comments or rationale concerning either the relative weighting of the children of a given node or the input attribute levels of the alternative systems. Thus, the amount of space for these comments is inversely proportional to the number of alternative systems. Also, because of this structure, the number of alternative systems is limited to 57. Finally, the objective associated with the given node is contained in the 64th to the 80th words.

III SENSITIVITY ANALYSIS

The sensitivity analyses are controlled by the routines in the primary overlay (4,0). The types of analysis handled by the program are: relative weight analysis (RSA), cumulative weight analysis (CSA), attribute level analysis (LSA), and a system analysis (SSA) based on one of the first three types. The first three types of analysis involve determining a fixed node and conducting the appropriate analysis for all systems at that node. The last type of analysis (actually three subtypes SSA/RSA, SSA/CSA, SSA/LSA) involves fixing the alternative system and then conducting the appropriate analysis over a set of nodes.

The entry point of the overlay, routine SENSTV is used to determine the main type of analysis (CSA, RSA, LSA, or SSA). For any of the first three types of sensitivity analysis, control is shifted to routine DETNOD which is used to fix the analysis to the desired node. Routine DETNOD also establishes the range over which the factor (cumulative weight, relative weight, or attribute level) is to be considered. For the last type of analysis (SSA), control shifts to routine SYSEN which in turn calls upon the routines, SENVAL, SENTYP, and NODSET to establish the fixed system, the factor to be considered, and the set of nodes over which the analysis is to be conducted.

Regardless of the type of analysis to be performed, once the initialization process (as described in the preceding paragraph), has been completed, control shifts to the routine COMPUT. This routine utilizes the equations presented in Chapter IV (of Volume I) to generate a matrix of overall values. Each row of this matrix corresponds to a different level of the factor under consideration. For the CSA, RSA, and LSA, each column

corresponds to a different system. For the SSA, each column represents a different node.

After the overall values have been calculated, the routines GRAPH and TABDIS are used to display the information in a graphical or tabular manner respectively (see Figures 4.1 through 4.4 in Volume I). At the conclusions of the desired type of sensitivity analysis, control is turned over to the main option selection.

IV HIERARCHY MANIPULATION

Subroutines NODIN and FIND

These subroutines are used to allow the user to enter the tree structure at a desired node. If the desired node is not an element of the tree, FIND will identify that node which is closest to the input node. All nodes are referenced by their node reference number (NRN). Except for option *****MOD*****, the input of a non-validated NRN will be interpreted as an indication that the user desires to exit from the option. Program control will be return to the calling portion of MADAM. For option *****MOD***** an invalide NRN is interpreted as a desire to introduce a new node having the input NRN, so the appropriate branches (and dummy connecting nodes) are constructed so as to create a valid environment for the input node.

Subroutine NODIN. This routine allows the user to input a desired NRN. If anything but a null string (space followed by <CR>) is entered, this routine will call on FIND to establish the existence of a node corresponding to the input NRN.

Subroutine FIND. This routine implements a modified breadth-first search of the tree to search for a node corresponding to the input NRN. With each digit of the NRN representing a level, FIND starts with the first digit and attempts to locate a node with the desired NRN digit by crossing the span at that level. This process is repeated for all remaining levels (digits) of the input NRN. If a match cannot be established at a given level, this indicates that the desired node would be an added crosslink at that level. If a match is established, the next digit is brought up for consideration. If no more downlink path is available (a dat node has been reached) the routine is terminated.

Figure 2 illustrates an example of this search process.

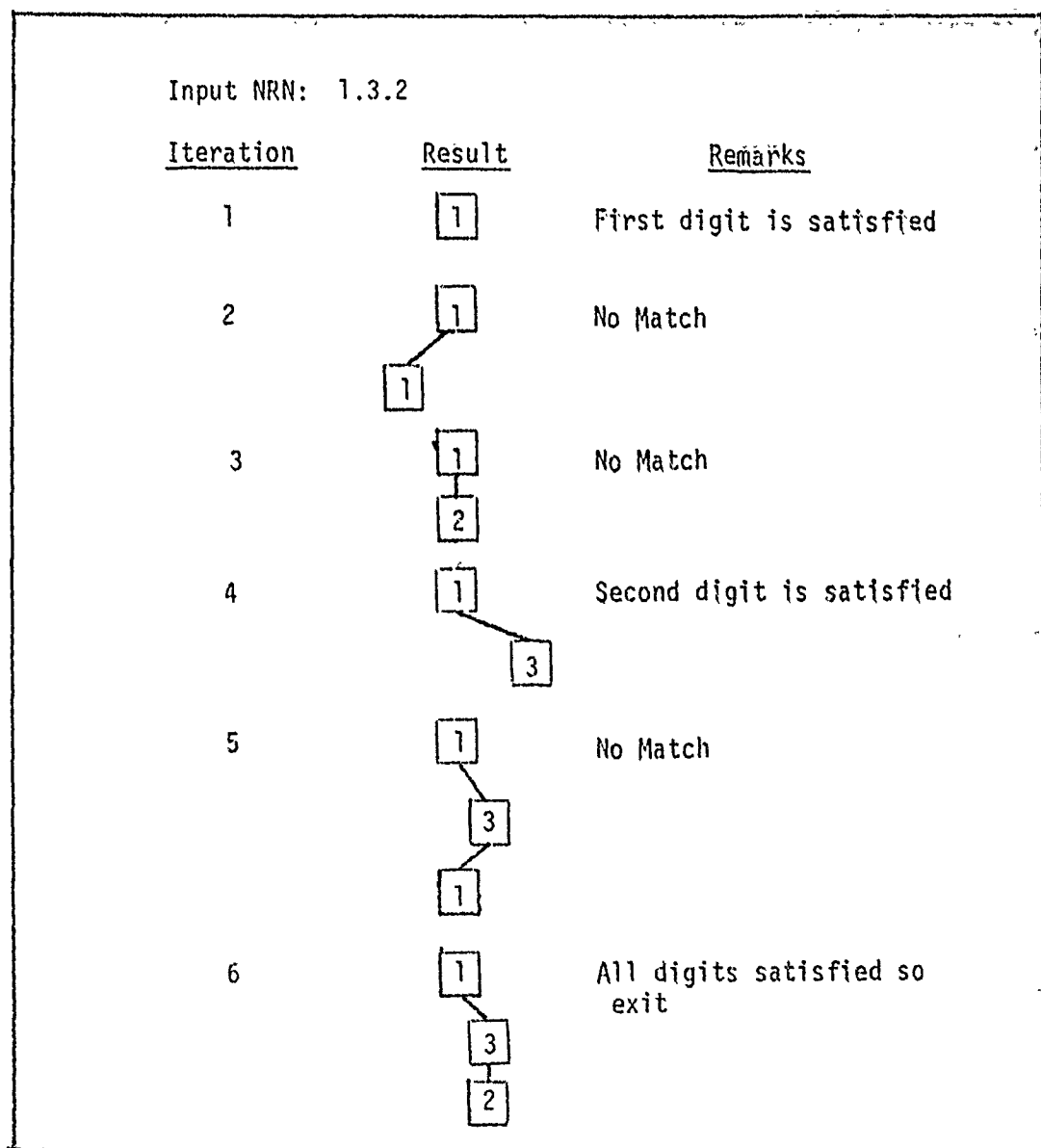


Figure 2 The Search Process (FIND)

Subroutines PRENEX, PRETOT, and NEXT

These routines drive a depth-first search of the tree structure. Subroutines PRENEX and PRETOT are used to initialize the search process at a desired node or at the root node respectively. Once all variables have been initialized, subroutine NEXT performs a depth-first traversal, advancing one node each time it is called.

Subroutine PRENEX. This routine elicits a NRN from the user by invoking NODIN. If a valid NRN is entered, the traversal will begin at the node corresponding to the input NRN. All descendants of this input node will be encountered in the resulting transversal.

Subroutine PRETOT. This routine initialized the variables required by NEXT so as to run a transversal of the entire tree structure. It is usually invoked just prior to a potential application of PRENEX, thus allowing the user to gracefully opt a tour of the entire hierarchical structure.

Subroutine NEXT. This routine tours the hierarchical structure beginning at the node specified by either PRETOT or PRENEX. Since a check is made as to the validity of the input node before any processing is done, it is possible to exit the current option by entering a null string during option PRENEX. The logic implemented in NEXT is illustrated in Figure 3.

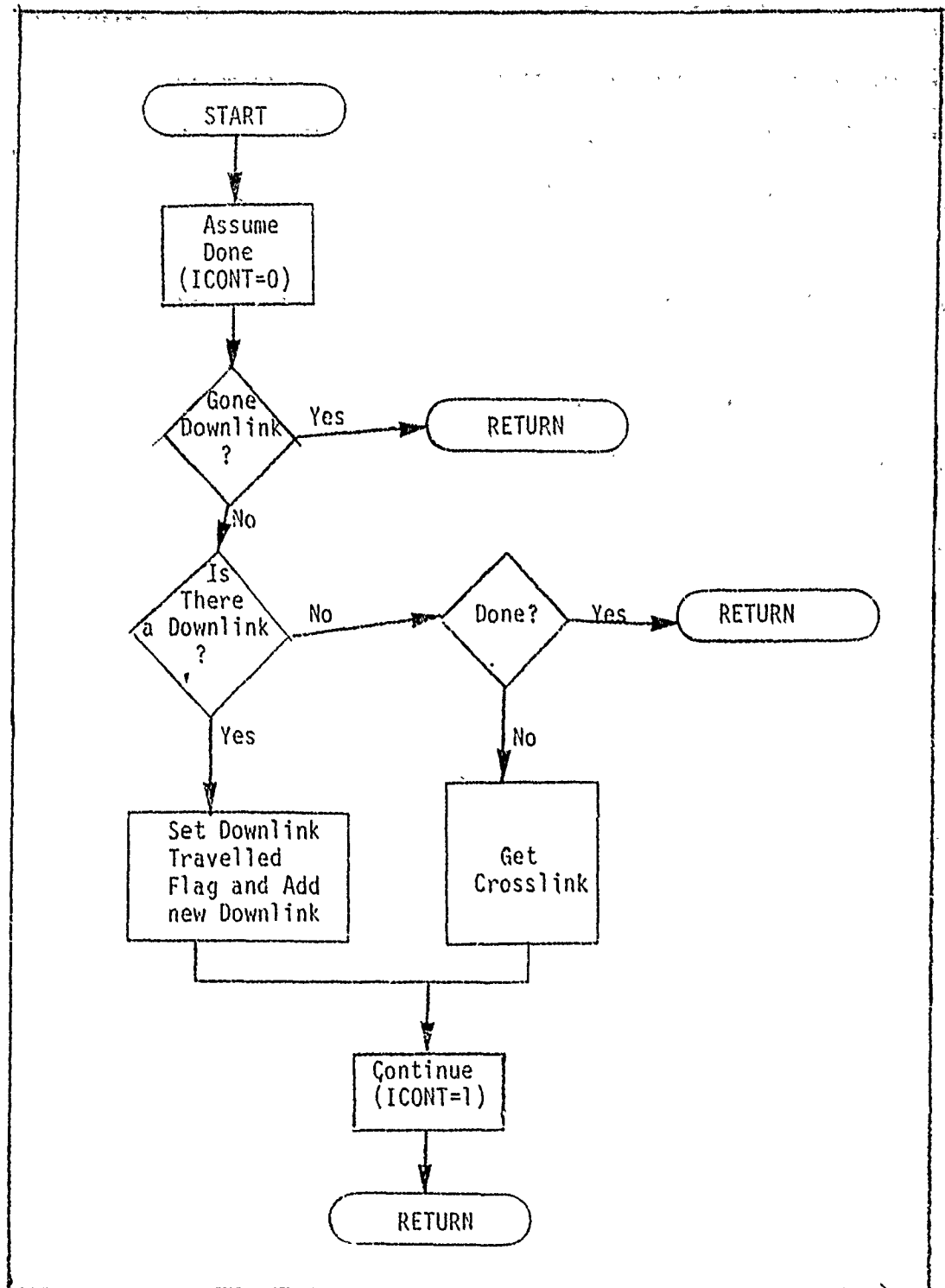


Figure 3 Logic Flow for NEXT

Subroutine SPAN

This routine allows for the input of the objective hierarchy structure by sequentially providing nodes in a depth-first fashion. The user inputs immediate descendants of the current node. Subroutine CHECK is used to establish a satisfactory set of children for the current node. Once a satisfactory set is established the first descendant of the current node becomes the new parent node. When a data node is reached (no descendants are input), the crosslink to the current node becomes the new parent node. If no more siblings exist, the routine will back up one level and establish the first crosslink as the new parent node (see Figure in the main thesis text).

Subroutine CALC

This routine calculates intermediate values (collapses the tree) by implementing a modified depth-first search. The tree is examined by each path from the root node to a data node, and the cumulative weights are calculated. System values based on the attributes are then computed for each node. The logic flow of CALC is provided in Figure 4. The cumulative weight of each node is given by:

$$\text{CUMWT (NODE)} = \text{CUMWT(PARENT)} * \text{RELWT(NODE)}$$

The system value at a particular node are the products of the system value at a data node and the cumulative weight of the upper nodes for each system.

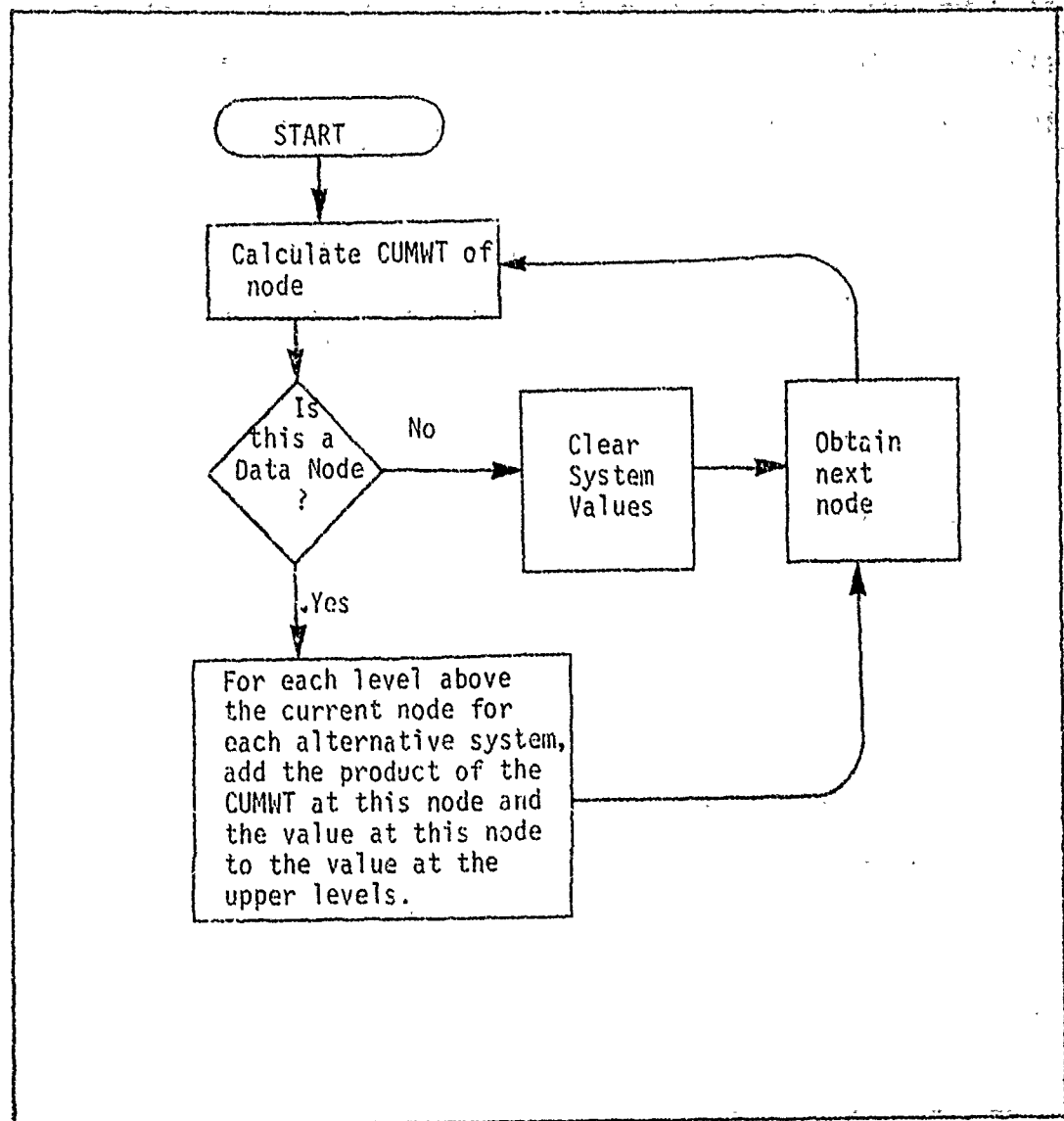


Figure 4 Logic Flow of CALC

V. Variable List

Variables Used

The following is a list of variables used in the program. This list includes only important variables and does not include scratch variables, do-loop indices, and dummy arguments. Psuedo-array variables are in the next section.

AIN - vector of relative weights

ARRAY - used to read and write to the data file. It holds one record worth of information

ATTMAX - maximum level of attribute

ATTMIN - minimum level of attribute

ATTNAM - name of attribute being perturbed during sensitivity analysis

ATTR1 - name of first attribute

ATTR2 - name of second attribute

B0 - intercept of linear regression (linear form)

BOEXP - intercept of linear regression (exponential form)

BOLN - intercept of linear regression (logarithmic form)

BOROOT - intercept of linear regression (square-root form)

BOSQR - intercept of linear regression (squared form)

B1 - slope coefficient of linear regression (linear form)

B1EXP - slope coefficient of linear regression (exponential form)

B1LN - slope coefficient of linear regression (logarithmic form)

B1ROOT - slope coefficient of linear regression (square-root form)

B1SQR - slope coefficient of linear regression (squared form)

BAND - boundaries of indifference curve window

CMD - main option selection

DEL1 - change in first attribute
 DEL2 - change in second attribute
 ERRFLG - counts the number of consecutive errors in user entry during main option selection
 FACLVL - level of attribute
 FINDST - set of IFIND for SSA
 ICHECK - flag if using between-node check
 0 - not using check
 1 - using check
 ICONT - flags whether or not to continue processing
 0 - stop
 1 - continue
 IDATA - flags whether or not node is data node
 0 - interior node
 1 - data node
 IFADD - flags relationship of current node to input node. Determines how to add the input branch to the existing branch.
 2 - IFIND is parent to input node
 3 - IFIND is sibling to input node
 IFIND - points to the last node of the branch contained in INNRN
 INDERR - flag for PPI
 0 - PPI hold
 1 - PPI fails
 INDX - NRN digit for IFIND
 INDXST - set of INDX for SSA
 IQUIT - flag to continue processing
 0 - continue
 1 - stop
 IRRAY - used to read and write to the data file. It holds one record worth of information.
 ISAVD - flags whether current information is saved on data file in case of abort
 0 - not saved
 1 - saved
 ISIN - record number of record currently in memory (in ARRAY or IRRAY)

ISYS - number of systems being considered during sensitivity analysis
 ITOL - tolerance during PPI testing
 ITOTL - flags type of tree traversal
 0 - not a total tree traversal
 1 - traverse all descendants
 KIDSET - set of NCHILD for nodes during SSA
 LEVEL - vector containing branch to node IFIND
 (*,1) - location of record containing node *
 (*,2) - flags whether or not visited by PRETOT-PRENEX-NEXT
 (*,3) - used in copying
 LINE - one line of data for plot
 LVL - length of LEVEL vector
 MATRIX - pairwise-comparison matrix
 MAX1 - maximum level of first attribute
 MAX2 - maximum level of second attribute
 MIN1 - minimum level of first attribute
 MIN2 - minimum level of second attribute
 MTYPE - type of SSA to be done
 NATT - number of attributes
 NCHILD - number of children for a node
 NCROSS - number of siblings on a span
 NDEEP - depth of tree (maximum number of levels)
 NDIFF - number of digits in INNRN not matched by routine IFIND
 NLOUD - number of allowed nodes plus four
 NLVLS - length of INNRN
 NNODES - number in tree plus one (master)
 NSYS - number of alternative systems
 NTAPE - defines which data files is in use
 NTREE - maximum number of simultaneous data files

OBJECTV -- objective associated with a particular node
 SENS - type of sensitivity analysis
 SSE - sum of squared error (linear form)
 SSEEXP - sum of squared error (exponential form)
 SSELN - sum of squared error (logarithmic form)
 SSERT - sum of squared error (square-root form)
 SSESQR - sum of squared error (squared form)
 TOLER - tolerance for PPI testing
 TOP - objective of parent node during between-node check
 USER - user identifier
 VAL - attribute levels which define data points for the individual value function
 WHOLD - matrix of overall value of systems during sensitivity analysis
 WMAX - maximum level of factor during sensitivity analysis
 WMIN - minimum level of factor during sensitivity analysis

Pseudo-array Variables

The following are functions which are treated as if they were array variables (see Table 1).

ARRAY(I,J)	relative and cumulative weight of node
ATT(I,J)	best/worst levels of attributes
ATT1(I)	attribute names
CRAY(I)	comments or rationale (subroutine)
IRAY(I,J)	node position information
LABEL(I)	name of attribute associated with node I
OBJECT(I,OBJECTV)	objective at node I
PARAM(I,J)	regression parameters

SYSLBL(I) alternative system names

VRAY(I,J) system values at node I

Cross Reference Map

The listing contains a partial load map, and combined with the comments within each program unit, a cross reference map of where each variable is used or modified may be generated.

VI. Program Structure

MADAM is divided up into five overlays (one main, four primary) which reduce the required field length for loading to about 60K. These overlays are machine dependent, but they may be removed by deleting the overlay statements, changing the program statement following each overlay statement to a subroutine statement (except for PROGRAM MADAM), and changing all overlay calls to subroutine calls to the subroutine making entry into the described overlay. Each primary overlay is designed to operate independently of the other primary overlays. Only the main overlay and one primary overlay are in the user's field length at one time, thus considerably reducing the required core memory allocation. The following is a list of routines in MADAM in the order in which they appear in the source code.

```
OVERLAY (XFILE,0,0) - main overlay
PROGRAM MADAM
  FUNCTION ARAY(I,J)
  SUBROUTINE ASET(I,J,V)
  FUNCTION ATT(I,J)
  FUNCTION ATT1(I)
  SUBROUTINE ATTLBL(I,ATTRIB)
  SUBROUTINE ATTSET(I,ATTRIB)
  SUBROUTINE ATTVAL(I,J,V)
  SUBROUTINE COPYR
  SUBROUTINE CRAY(I)
  SUBROUTINE CSET(I)
  SUBROUTINE FIND
  SUBROUTINE HELP
  SUBROUTINE INTRO
  FUNCTION IRAY(I,J)
  SUBROUTINE ISET(I,S,M)
  FUNCTION LABEL(I)
  SUBROUTINE LSET(I,LABELT)
  SUBROUTINE NEXT
  SUBROUTINE NODIN
  SUBROUTINE OBJECT(I,OBJECTV)
  SUBROUTINE OSET(I,OBJECTV)
  FUNCTION PARAM(I,J)
```

```

SUBROUTINE PRENEX
SUBROUTINE PRETOT
SUBROUTINE PSET(I,J,V)
FUNCTION SYSLBL(I)
SUBROUTINE SYSSET(I,SYSLBL)
SUBROUTINE TLOAD
SUBROUTINE TSAVE
FUNCTION VALU(X,LABEL)
FUNCTION VRAY(I,J)
SUBROUTINE VSET(I,K,V)
FUNCTION XLEVEL(X,LABEL)

```

OVERLAY (XFILE,1,0) - primary overlay
PROGRAM DUMMY

```

SUBROUTINE ADDSYS
SUBROUTINE ASK(I, X1,X2,XMID)
SUBROUTINE CHECK
SUBROUTINE DELSYS
SUBROUTINE GRAFIX
SUBROUTINE INITT
SUBROUTINE MASTER
SUBROUTINE MODIFY
SUBROUTINE NEW
SUBROUTINE PICTUR(INDEX)
SUBROUTINE PPI
SUBROUTINE PRUNE
SUBROUTINE RDATT
SUBROUTINE RDSYSL
SUBROUTINE RDTTL
SUBROUTINE REGRS(INDEX)
SUBROUTINE SPAN
SUBROUTINE STAT
SUBROUTINE TAPER
SUBROUTINE VALUE

```

OVERLAY (XFILE,2,0) - primary overlay
PROGRAM WVLOAD

```

SUBROUTINE CALL
SUBROUTINE DVIDE(N,A,D)
SUBROUTINE GMVEC(NCROSS,IFIND,AIN)
SUBROUTINE RDV
SUBROUTINE RDWT
FUNCTION SUM(N,A)
SUBROUTINE NVLODI(OPT)

```

OVERLAY (XFILE,3,0) - primary overlay
PROGRAM NUM

```

SUBROUTINE DISPLA
SUBROUTINE DSPLOT
SUBROUTINE NEWPG
SUBROUTINE NUMREV

```

OVERLAY (XFILE,4,0) - primary overlay
PROGRAM SENSTV
SUBROUTINE COMPUT
SUBROUTINE DETNOD
SUBROUTINE HEADER(I,OBJECTV,ANSWER)
SUBROUTINE SENVAL
SUBROUTINE TABDIS

SOURCE CODE OF MADAM

Partial Load Map

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
C      OVERLAY(XFILE,0,0)
C      PROGRAM MADAN
C=====
C1 THIS PROGRAM IS THE MAIN PROGRAM OF MADAN. IT IS USED TO
C2 CONTROL THE MAIN OPTION SELECTION.
C3
C4 CALLED BY: (NONE)
C5
C6 VARIABLES
C7 USED: (NONE)
C8
C9 MODIFIED: CMD,ERRFLG
C=====
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C34
C35
C36
C37
C38
C39
C40
C41
C42
C43
C44
C45
C46
C47
C48
C49
C50
C51
C52
C53
C54
C55
C56
C=====
C10 COMMON/ATTR / HATT
C11 COMMON/C/NODES,NDEEP
C12 COMMON/CTRL/ISAUD,NTAPE
C13 COMMON/LEVEL/MLULS,INRM(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
C14 COMMON/NE/ICONT,IMATA,ITOTL
C15 COMMON/ONE/CMD
C16 COMMON/P/USER
C17 COMMON/RRAY/ARRAY(88)
C18 COMMON/RRAYE/ISIN,INLOUD
C19 COMMON/SYS/NSYS
C20 SAVE
C21 CHARACTER*3 CMD,USER*10
C22 INTEGER ERRFLG
C23 CALL INTRO
C24 NTAPE=1
C25 NSYS=2
C26 NDEEP=0
C27 CMD='SEL'
C28 CALL TLOAD
C29 ERRFLG=0
C30 WRITE(6, '(S(//),IX)')
C31 PRINT*,USER,'YOUR OPTIONS ARE:'
C32 PRINT*,ATT CDP DIS DON MOD NEW NUM PRU REV SEL'
C33 PRINT*,SEN SPA STA SYS TTL WJC'
C34 PRINT*,XWRITE: IF YOU NEED AN E'PLAMATION, 'USER
C35 PRINT*,TYPE 'HELP' XXX'
C36 PRINT*,
C37 PRINT*,WHAT IS YOUR CHOICE. 'USER,'?
C38 READ(1, '(A3),XCHD
C39 IF(CMD.EQ.'ATT')THEN
C40 ERRFLG=0
C41 CALL OVERLAY(XFILE,1,0)
C42
C43 7 CHARACTER SYMBOL _OVERLAY IS NON- ANGI
C44 ELSEIF(CMD.EQ.'CDP')THEN
C45 ERRFLG=0

```

```

57 CALL COPYR
58 ELSE IF (CMD.EQ.'DIS') THEN
59   ERRFLG=0
60   CALL OVERLAY('XFILE',3,0)
61   ELSE IF (CMD.EQ.'DEL') THEN
62     ERRFLG=0
63     CALL HELP
64   ELSE IF (CMD.EQ.'MOD') THEN
65     ERRFLG=0
66   CALL OVERLAY('XFILE',1,0)
67   ELSE IF (CMD.EQ.'REV') THEN
68     ERRFLG=0
69   CALL OVERLAY('XFILE',1,0)
70   ELSE IF (CMD.EQ.'NUM') THEN
71     ERRFLG=0
72   CALL OVERLAY('XFILE',3,0)
73   ELSE IF (CMD.EQ.'PRU') THEN
74     ERRFLG=0
75   CALL OVERLAY('XFILE',1,0)
76   ELSE IF (CMD.EQ.'REV') THEN
77     ERRFLG=0
78   CALL OVERLAY('XFILE',3,0)
79   ELSE IF (CMD.EQ.'SEL') THEN
80     ERRFLG=0
81   CALL OVERLAY('XFILE',1,0)
82   ELSE IF (CMD.EQ.'SEN') THEN
83     ERRFLG=0
84   CALL OVERLAY('XFILE',4,0)
85   ELSE IF (CMD.EQ.'SPA') THEN
86     ERRFLG=0
87   CALL OVERLAY('XFILE',1,0)
88   ELSE IF (CMD.EQ.'STA') THEN
89     ERRFLG=0
90   CALL OVERLAY('XFILE',1,0)
91   ELSE IF (CMD.EQ.'SYS') THEN
92     ERRFLG=0
93   CALL OVERLAY('XFILE',1,0)
94   ELSE IF (CMD.EQ.'TTL') THEN
95     ERRFLG=0
96   CALL OVERLAY('XFILE',1,0)
97   ELSE IF (CMD.EQ.'UBC') THEN
98     ERRFLG=0
99   CALL OVERLAY('XFILE',2,0)
100   ELSE
101     ERRFLG=ERRFLG+1
102   IF (CMD.NE.'DON') PRINT*,CMD,' IS NOT AN ALLOWED ENTRY. 'USER,'
103   ENDIF
104   IF (ERRFLG.GE.3).AND.(CMD.NE.'DON')) THEN
105     PRINT*, 'TO MINIMIZE ENTRY ERRORS, AN EXPLANATION WILL BE PROVIDED'
106     CALL HELP
107     ERRFLG=0
108     ENDIF
109   IF (CMD.NE.'DON') GO TO 1
110   CALL TSAME
111   STOP
112   END

```

[illegible]

```

--PROCEDURES--(loop)
NAME      TYPE      ARG3      CLASS      NAME      TYPE      ARG3      CLASS
-----
COPYR      0      SUBROUTINE
HALP      0      SUBROUTINE
INTRO      0      SUBROUTINE
OVERLAY    3      SUBROUTINE
TLOAD      0      SUBROUTINE
TSAVE      0      SUBROUTINE

```

```

- STATEMENT LABELS--(LO-A)
- LABEL-ADDRESSES---PROPERTIES---DEF
1      CR      42

```

---ENTRY POINTS---(L.O.A)
---PAGE---AUDITS---(A.O.A)
2010M 48 0

```

--STATISTICS--
PROGRAM-UNIT LENGTH      82      0
CM LABELLED COMMON LENGTH 2648    188
CM STORAGE USED          662962    24860
COMPILE TIME             0.109 SECONDS

```

2

—

4. බලාපොරොත්තු වන්නේ ආයතනිකව පැවැත්වෙන සම්මන්ත්‍රණයක් මගින් විශ්ලේෂණය කළ යුතු බවයි.

ကမ္ဘာ့ဆရာတော်များကလည်း
လူလူလူလူကကကကကက

ကမ္ဘာ့ဆရာတော်တို့၏
ပညာပညာပညာပညာ

2014年12月15日

၆၆၆
 ၆၆၆

සංස්කෘත

3333

৯৯

၁၂၂၂

35

ABT

1

4

On

615

4

8

ABT

1

4

On

615

4

8

87

```

SUBROUTINE ASET(I,J,U)
C-----
C THIS ROUTINE IS USED TO SET THE VALUES OF THE PSEUDO-ARRAY
C VARIABLE ARRAY(I,J). ASET(2,I,U) SETS THE RELATIVE HEIGHT
C OF NODE I TO U. ASET(3,2,U) SETS THE CURVATURE HEIGHT
C OF NODE 3 TO U. THE VALUES ARE ACCESSED VIA FUNCTION
C ARRAY(I,J).
C-----
      CALLED BY: MODIFY, SPAN, CALC, RDOT
C-----
      VARIABLES
        USED: I,J,NTAPE,U
        MODIFIED: APRAY,ISIN,K
C-----
      COMMON/CNTRL/ISAUD,NTAPE
      COMMON/SRVL/APRAY(80)
      COMMON/SRVE/ISIN,KLOD
      IF(.NE.ISIN)READ(NTAPE,REC=1,ERR=99)(APRAY(K),K=1,20)
      APRAY(J+4)=U
      WRITE(NTAPE,REC=1,ERR=99)(APRAY(K),K=1,20)
      ISIN=I
      RETURN
      PRINT*, '**O ERROR IN ASET'
      RETURN
      END
C-----
      99
C-----

```

67B = 55
124B = 84
6660B = 24000
0.030 SECONDS

--VARIABLE MAP--(LO-A)	--NAME--ADDRESS--BLOCK----	--PROPERTIES--TYPE-----SIZE	--NAME--ADDRESS--BLOCK----	--PROPERTIES--TYPE-----SIZE
APRAY	08	SRVL/1/	K	G4P
I	01	DURNV-ARG	KLOD	18
ISAUD	03	CNTRL/	NTAPE	18
ISIN	08	/ARRAY/	U	3
J	02	DURNV-ARG		

SUBROUTINE ASET
 --ENTRY POINTS--(LO-A)
 --NAME--ADDRESS--ARQ---
 ASET 5B 3
 --STATISTICS--
 PROGRAM-EXIT LENGTH
 CN LABELLED COMMON LENGTH
 CN STORAGE USED
 COMPILE TIME

```

---STATEMENT LABELS---(LO-P)
-LABEL-ADDRESS-----PROPERTIES---DEF
      98      348      34

```


[illegible]

```

1 SUBROUTINE ACTUAL(I,J,U)
2 COMMON/CTRL/ISAUD,NTAPE
3 COMMON/ARRAY/ARRAY(88)
4 COMMON/ARRAY2/ISIN,NLOUD
5 SAVE
6 K=NLOUD*6+J
7 IF(K.NE.ISIN)READ(NTAPE,REC=K,ERR=99)(ARRAY(L),L=1,88)
8 ARRAY(I)=C
9 WRITE(NTAPE,REC=K,ERR=99)(ARRAY(L),L=1,88)
10 ISIN=K
11 RETURN
12 PRINT*, 'I/O ERROR IN ACTUAL'
13 RETURN
14 END

```

```

--VARIABLE LABELS--(LO=A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
K 18 /SEASUSE/ SRU INTEGER
L 28 /SEASUSE/ SRU INTEGER
NLOUD 18 /RRAY2/ INTEGER
NTAPE 18 /CTRL/ INTEGER
U 3 DUMMY-ARG REAL

```

```

--STATEMENT LABELS--(LO=A)
--LABEL-RADR(S)---PROPERTIES---DEF
99 378 12

```

```

--ENTRY POINTS--(LO=A)
--NAME--ADDRESS--ARGS--
ACTUAL 52 3

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH 708 55
CT LABELLED COMMON LENGTH 1278 87
CO STORAGE USED 606008 24968
COMPILE TIME 0.037 SECONDS

```

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE COPYR
C*****
C THIS ROUTINE IS USED TO COPY THE CHILDREN OF ONE NODE
C TO ANOTHER NODE. THE OBJECTIVE AND WEIGHTS ARE
C TRANSFERRED.
C*****
C CALLED BY: RADAK
C*****
C VARIABLES
C USED: ICORT, IDATA, IFADD, IFIND, NDIFF, NLULS
C MODIFIED: IFINDT, LEVEL, NNODES
C*****
COMMON/C/NODES, NDEEP
COMMON/C/INTL/ISAD, NTRAPE
COMMON/LE/ST/NLULS, IRRAY(20), IFIND, NDIFF, IFADD, LUL, LEVEL(20,3)
COMMON/NEK/ICORT, IDATA, ITOTL
COMMON/STAV1/IRRAY(20)
COMMON/RRAY2/ISIM, NLOUD
SAVE
CHARACTER*10 OBJECTU(10)
PRINT*, 'ENTER NODE TO BE COPIED TO.'
CALL MODIN
IF (NLULS.LE.0) RETURN
IF (NDIFF.GT.0) THEN
PRINT*, 'MODE MUST EXIST BEFORE COPYING IS ALLOWED.'
RETURN
ENDIF
IFINDT=IFIND
PRINT*, 'ENTER NODE TO BE COPIED.'
CALL PRENEX
IF (ICORT.EQ.0) RETURN
IF (NDIFF.GT.0) THEN
PRINT*, 'MODE TO BE COPIED DOES NOT EXIST.'
RETURN
ENDIF
LEVEL(LUL,3)=IFINDT
IF (IDATA.NE.0) THEN
PRINT*, 'THE NODE ENTERED IS A DATA NODE.'
PRINT*, 'AND CANNOT BE COPIED FROM.'
RETURN
ENDIF
CALL NEXT
IF (ICORT.EQ.0) THEN
PRINT*, 'COPYING IS COMPLETED.'
RETURN

```



```

123  SUBROUTINE CRAY(I)
124  C
125  C*****
126  C THIS ROUTINE IS USED TO ACCESS COMMENTS (OTHER THAN THE
127  C OBJECTIVE) ASSOCIATED WITH A MODE. THESE COMMENTS WILL
128  C BE A TITLE FOR THE DATA STRUCTURE FOR CRAY(I). THEY WILL
129  C BE RATIONALE FOR THE RELATIVE WEIGHTS OF THE CHILDREN FOR
130  C AN INTERIOR (NON-DATA) NODE. THEY WILL BE COMMENTS
131  C CONCERNING ATTRIBUTE LEVELS FOR A DATA MODE. THESE
132  C COMMENTS ARE ENTERED VIA SUBROUTINE CSET(I).
133  C
134  C CALLED BY: NEUPG,MURREU
135  C
136  C VARIABLES
137  C   USED: I,MSYS,NTAPE
138  C   MODIFIED: ARRAY,ISIM,ISTART,ISTOP,J,K
139  C
140  C*****
141  C
142  C COMMON/CTRL/ISAUD,NTAPE
143  C COMMON/ARRAY/ARRAY(80)
144  C COMMON/ARRAY2/ISIN,NLOD
145  C COMMON/SYS1/MSYS
146  C SAVE
147  C CHARACTER*10 ARRAY
148  C ISTART=8+MSYS
149  C IF (ISIN.NE.I) READ(NTAPE,REC=1,ERR=92)((ARRAY(K),K=1,80)
150  C IF (ARRAY(ISTART).EQ.'') THEN
151  C   ISIN=I
152  C   RETURN
153  C ENDIF
154  C ISTOP=ISTART+7
155  C IF (ISTOP.GT.64) THEN
156  C   ISIN=I
157  C   RETURN
158  C ENDIF
159  C WRITE(3,'(1X,80)')(ARRAY(J),J=ISTART,ISTOP)
160  C ISTART=ISTART+8
161  C GO TO 1
162  C PRINT2,'I/O ERROR IN CRAY'
163  C RETURN
164  C END
165  C
166  C
167  C
168  C
169  C
170  C
171  C
172  C
173  C
174  C
175  C
176  C
177  C
178  C
179  C
180  C
181  C
182  C
183  C
184  C
185  C
186  C
187  C
188  C
189  C
190  C
191  C
192  C
193  C
194  C
195  C
196  C
197  C
198  C
199  C
200  C
201  C
202  C
203  C
204  C
205  C
206  C
207  C
208  C
209  C
210  C
211  C
212  C
213  C
214  C
215  C
216  C
217  C
218  C
219  C
220  C
221  C
222  C
223  C
224  C
225  C
226  C
227  C
228  C
229  C
230  C
231  C
232  C
233  C
234  C
235  C
236  C
237  C
238  C
239  C
240  C
241  C
242  C
243  C
244  C
245  C
246  C
247  C
248  C
249  C
250  C
251  C
252  C
253  C
254  C
255  C
256  C
257  C
258  C
259  C
260  C
261  C
262  C
263  C
264  C
265  C
266  C
267  C
268  C
269  C
270  C
271  C
272  C
273  C
274  C
275  C
276  C
277  C
278  C
279  C
280  C
281  C
282  C
283  C
284  C
285  C
286  C
287  C
288  C
289  C
290  C
291  C
292  C
293  C
294  C
295  C
296  C
297  C
298  C
299  C
300  C
301  C
302  C
303  C
304  C
305  C
306  C
307  C
308  C
309  C
310  C
311  C
312  C
313  C
314  C
315  C
316  C
317  C
318  C
319  C
320  C
321  C
322  C
323  C
324  C
325  C
326  C
327  C
328  C
329  C
330  C
331  C
332  C
333  C
334  C
335  C
336  C
337  C
338  C
339  C
340  C
341  C
342  C
343  C
344  C
345  C
346  C
347  C
348  C
349  C
350  C
351  C
352  C
353  C
354  C
355  C
356  C
357  C
358  C
359  C
360  C
361  C
362  C
363  C
364  C
365  C
366  C
367  C
368  C
369  C
370  C
371  C
372  C
373  C
374  C
375  C
376  C
377  C
378  C
379  C
380  C
381  C
382  C
383  C
384  C
385  C
386  C
387  C
388  C
389  C
390  C
391  C
392  C
393  C
394  C
395  C
396  C
397  C
398  C
399  C
400  C
401  C
402  C
403  C
404  C
405  C
406  C
407  C
408  C
409  C
410  C
411  C
412  C
413  C
414  C
415  C
416  C
417  C
418  C
419  C
420  C
421  C
422  C
423  C
424  C
425  C
426  C
427  C
428  C
429  C
430  C
431  C
432  C
433  C
434  C
435  C
436  C
437  C
438  C
439  C
440  C
441  C
442  C
443  C
444  C
445  C
446  C
447  C
448  C
449  C
450  C
451  C
452  C
453  C
454  C
455  C
456  C
457  C
458  C
459  C
460  C
461  C
462  C
463  C
464  C
465  C
466  C
467  C
468  C
469  C
470  C
471  C
472  C
473  C
474  C
475  C
476  C
477  C
478  C
479  C
480  C
481  C
482  C
483  C
484  C
485  C
486  C
487  C
488  C
489  C
490  C
491  C
492  C
493  C
494  C
495  C
496  C
497  C
498  C
499  C
500  C
501  C
502  C
503  C
504  C
505  C
506  C
507  C
508  C
509  C
510  C
511  C
512  C
513  C
514  C
515  C
516  C
517  C
518  C
519  C
520  C
521  C
522  C
523  C
524  C
525  C
526  C
527  C
528  C
529  C
530  C
531  C
532  C
533  C
534  C
535  C
536  C
537  C
538  C
539  C
540  C
541  C
542  C
543  C
544  C
545  C
546  C
547  C
548  C
549  C
550  C
551  C
552  C
553  C
554  C
555  C
556  C
557  C
558  C
559  C
560  C
561  C
562  C
563  C
564  C
565  C
566  C
567  C
568  C
569  C
570  C
571  C
572  C
573  C
574  C
575  C
576  C
577  C
578  C
579  C
580  C
581  C
582  C
583  C
584  C
585  C
586  C
587  C
588  C
589  C
590  C
591  C
592  C
593  C
594  C
595  C
596  C
597  C
598  C
599  C
600  C
601  C
602  C
603  C
604  C
605  C
606  C
607  C
608  C
609  C
610  C
611  C
612  C
613  C
614  C
615  C
616  C
617  C
618  C
619  C
620  C
621  C
622  C
623  C
624  C
625  C
626  C
627  C
628  C
629  C
630  C
631  C
632  C
633  C
634  C
635  C
636  C
637  C
638  C
639  C
640  C
641  C
642  C
643  C
644  C
645  C
646  C
647  C
648  C
649  C
650  C
651  C
652  C
653  C
654  C
655  C
656  C
657  C
658  C
659  C
660  C
661  C
662  C
663  C
664  C
665  C
666  C
667  C
668  C
669  C
670  C
671  C
672  C
673  C
674  C
675  C
676  C
677  C
678  C
679  C
680  C
681  C
682  C
683  C
684  C
685  C
686  C
687  C
688  C
689  C
690  C
691  C
692  C
693  C
694  C
695  C
696  C
697  C
698  C
699  C
700  C
701  C
702  C
703  C
704  C
705  C
706  C
707  C
708  C
709  C
710  C
711  C
712  C
713  C
714  C
715  C
716  C
717  C
718  C
719  C
720  C
721  C
722  C
723  C
724  C
725  C
726  C
727  C
728  C
729  C
730  C
731  C
732  C
733  C
734  C
735  C
736  C
737  C
738  C
739  C
740  C
741  C
742  C
743  C
744  C
745  C
746  C
747  C
748  C
749  C
750  C
751  C
752  C
753  C
754  C
755  C
756  C
757  C
758  C
759  C
760  C
761  C
762  C
763  C
764  C
765  C
766  C
767  C
768  C
769  C
770  C
771  C
772  C
773  C
774  C
775  C
776  C
777  C
778  C
779  C
780  C
781  C
782  C
783  C
784  C
785  C
786  C
787  C
788  C
789  C
790  C
791  C
792  C
793  C
794  C
795  C
796  C
797  C
798  C
799  C
800  C
801  C
802  C
803  C
804  C
805  C
806  C
807  C
808  C
809  C
810  C
811  C
812  C
813  C
814  C
815  C
816  C
817  C
818  C
819  C
820  C
821  C
822  C
823  C
824  C
825  C
826  C
827  C
828  C
829  C
830  C
831  C
832  C
833  C
834  C
835  C
836  C
837  C
838  C
839  C
840  C
841  C
842  C
843  C
844  C
845  C
846  C
847  C
848  C
849  C
850  C
851  C
852  C
853  C
854  C
855  C
856  C
857  C
858  C
859  C
860  C
861  C
862  C
863  C
864  C
865  C
866  C
867  C
868  C
869  C
870  C
871  C
872  C
873  C
874  C
875  C
876  C
877  C
878  C
879  C
880  C
881  C
882  C
883  C
884  C
885  C
886  C
887  C
888  C
889  C
890  C
891  C
892  C
893  C
894  C
895  C
896  C
897  C
898  C
899  C
900  C
901  C
902  C
903  C
904  C
905  C
906  C
907  C
908  C
909  C
910  C
911  C
912  C
913  C
914  C
915  C
916  C
917  C
918  C
919  C
920  C
921  C
922  C
923  C
924  C
925  C
926  C
927  C
928  C
929  C
930  C
931  C
932  C
933  C
934  C
935  C
936  C
937  C
938  C
939  C
940  C
941  C
942  C
943  C
944  C
945  C
946  C
947  C
948  C
949  C
950  C
951  C
952  C
953  C
954  C
955  C
956  C
957  C
958  C
959  C
960  C
961  C
962  C
963  C
964  C
965  C
966  C
967  C
968  C
969  C
970  C
971  C
972  C
973  C
974  C
975  C
976  C
977  C
978  C
979  C
980  C
981  C
982  C
983  C
984  C
985  C
986  C
987  C
988  C
989  C
990  C
991  C
992  C
993  C
994  C
995  C
996  C
997  C
998  C
999  C
1000 C

```

SUBROUTINE CRAY

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ARRAY	03	/ARRAY/		CHAR*10	50
ISAUD	1	DUMPY-ARG		INTEGER	
ISIN	68	/CNTRL/		INTEGER	
ISYST	68	/CNTRL/		INTEGER	
ISTOP	18	/SYS1/		INTEGER	
	38	/SYS1/		INTEGER	
		/CNTRL/		INTEGER	
		/SYS1/		INTEGER	
		/CNTRL/		INTEGER	
		/SYS1/		INTEGER	
		/CNTRL/		INTEGER	

--STATEMENT LABELS--(LO=A)
 --LABEL-ADDRESS-----PROPERTIES-----DEF
 1 22B 36
 98 73B 43

--ENTRY POINTS--(LO=A)
 --NAME--ADDRESS--ARGS--
 CRAY 58 1

--STATISTICS--
 PROGRAM-UNIT LENGTH 1378 : 95
 CN LABELLED COMMON LENGTH 1328 : 90
 CN STORAGE USED 606008 : 24950
 COMPILE TIME 0.061 SECONDS

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
SUBROUTINE CSET(I)
C
C *****
C THIS ROUTINE SETS THE COMMENTS ACCESSED BY SUBROUTINE
C CRAY(I). SEE CRAY(I) FOR A DESCRIPTION OF THEM.
C *****
C CALLED BY: RDTTL,RDU,RDUT
C *****
C VARIABLES
C   USED: I,NSYS,NTAPE
C   MODIFIED: ARRAY,ISIN,ISTART,ISTOP,J,K
C *****
C COMMON/CHTRL/ISAUD,NTAPE
C COMMON/ARRAY1/ARRAY(S0)
C COMMON/ARRAY2/ISIN,NILOUT
C COMMON/SYS1/NSYS
C
C CHARACTER*10 ARRAY
C ISTART=8
C IF(I.NE.ISIN)READ(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
C ISTOP=ISTART+7
C IF(ISTOP.GT.64)THEN
C   WRITE(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
C   ISIN=1
C   RETURN
C ENDIF
C PRINT*,''
C READ(1,100) (ARRAY(J),J=ISTART,ISTOP)
C IF(ARRAY(ISTART).EQ.'')THEN
C   WRITE(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
C   ISIN=1
C   RETURN
C ENDIF
C ISTART=ISTART+8
C GO TO 1
C PRINT*, 'I/O ERROR IN CSET'
C RETURN
C END

```

---VARIABLE MAP---(LO-A)				---VARIABLE MAP---(HI-A)			
NAME	ADDRESS	BLOCK	SIZE	NAME	ADDRESS	BLOCK	SIZE
RTNAY	00	RTNAY1/	20	CHRS10	40	CHRS10/	20
254UD	01	DURNV-ARG		INTEGER	41	SEASUW/	SAU
51H	02	CHTRL/		INTEGER	42	SEASUW/	SAU
151H	03	BRAY/		INTEGER	43	BRG2/	SAU
151H	04	SEASUW/	SAU	INTEGER	44	SEASUW/	SAU
151H	05	SEASUW/	SAU	INTEGER	45	SEASUW/	SAU
151H	06	SEASUW/	SAU	INTEGER	46	SEASUW/	SAU
151H	07	SEASUW/	SAU	INTEGER	47	SEASUW/	SAU
151H	08	SEASUW/	SAU	INTEGER	48	SEASUW/	SAU
151H	09	SEASUW/	SAU	INTEGER	49	SEASUW/	SAU
151H	10	SEASUW/	SAU	INTEGER	50	SEASUW/	SAU
151H	11	SEASUW/	SAU	INTEGER	51	SEASUW/	SAU
151H	12	SEASUW/	SAU	INTEGER	52	SEASUW/	SAU
151H	13	SEASUW/	SAU	INTEGER	53	SEASUW/	SAU
151H	14	SEASUW/	SAU	INTEGER	54	SEASUW/	SAU
151H	15	SEASUW/	SAU	INTEGER	55	SEASUW/	SAU
151H	16	SEASUW/	SAU	INTEGER	56	SEASUW/	SAU
151H	17	SEASUW/	SAU	INTEGER	57	SEASUW/	SAU
151H	18	SEASUW/	SAU	INTEGER	58	SEASUW/	SAU
151H	19	SEASUW/	SAU	INTEGER	59	SEASUW/	SAU
151H	20	SEASUW/	SAU	INTEGER	60	SEASUW/	SAU
151H	21	SEASUW/	SAU	INTEGER	61	SEASUW/	SAU
151H	22	SEASUW/	SAU	INTEGER	62	SEASUW/	SAU
151H	23	SEASUW/	SAU	INTEGER	63	SEASUW/	SAU
151H	24	SEASUW/	SAU	INTEGER	64	SEASUW/	SAU
151H	25	SEASUW/	SAU	INTEGER	65	SEASUW/	SAU
151H	26	SEASUW/	SAU	INTEGER	66	SEASUW/	SAU
151H	27	SEASUW/	SAU	INTEGER	67	SEASUW/	SAU
151H	28	SEASUW/	SAU	INTEGER	68	SEASUW/	SAU
151H	29	SEASUW/	SAU	INTEGER	69	SEASUW/	SAU
151H	30	SEASUW/	SAU	INTEGER	70	SEASUW/	SAU
151H	31	SEASUW/	SAU	INTEGER	71	SEASUW/	SAU
151H	32	SEASUW/	SAU	INTEGER	72	SEASUW/	SAU
151H	33	SEASUW/	SAU	INTEGER	73	SEASUW/	SAU
151H	34	SEASUW/	SAU	INTEGER	74	SEASUW/	SAU
151H	35	SEASUW/	SAU	INTEGER	75	SEASUW/	SAU
151H	36	SEASUW/	SAU	INTEGER	76	SEASUW/	SAU
151H	37	SEASUW/	SAU	INTEGER	77	SEASUW/	SAU
151H	38	SEASUW/	SAU	INTEGER	78	SEASUW/	SAU
151H	39	SEASUW/	SAU	INTEGER	79	SEASUW/	SAU
151H	40	SEASUW/	SAU	INTEGER	80	SEASUW/	SAU
151H	41	SEASUW/	SAU	INTEGER	81	SEASUW/	SAU
151H	42	SEASUW/	SAU	INTEGER	82	SEASUW/	SAU
151H	43	SEASUW/	SAU	INTEGER	83	SEASUW/	SAU
151H	44	SEASUW/	SAU	INTEGER	84	SEASUW/	SAU
151H	45	SEASUW/	SAU	INTEGER	85	SEASUW/	SAU
151H	46	SEASUW/	SAU	INTEGER	86	SEASUW/	SAU
151H	47	SEASUW/	SAU	INTEGER	87	SEASUW/	SAU
151H	48	SEASUW/	SAU	INTEGER	88	SEASUW/	SAU
151H	49	SEASUW/	SAU	INTEGER	89	SEASUW/	SAU
151H	50	SEASUW/	SAU	INTEGER	90	SEASUW/	SAU
151H							

-----PROPERTY-----DEF

94
16

9111
122

63
1

```
--ENTRY POINTS--(LO=0)
--NAME--ADDRESS--ARG5---
```

1550 58 1

```

PROGRAM-UNIT LENGTH      170B - 120
CM LABELLED COMMON LENGTH 122B - 50
CM STORAGE USED          60600B - 34960
COMPILE TIME             0.070 SECONDS

```


SUBROUTINE FIND

```

56      CONTINUE
57      IDATA=0
58      IF(IIRAY(IIFIND,2),LE,0)IDATA=1
59      RETURN
60      END
61
62

```

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
YCONT	0B	/NEX/		INTEGER	
IDATA	1B	/NEX/		INTEGER	
IFADD	27B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INMAY	1B	/LEVEL/		INTEGER	
ITOTL	2B	/NEX/		INTEGER	
J	LEVEL			INTEGER	60
LVL	21B	/LEVEL/		INTEGER	
NDIFF	20B	/LEVEL/		INTEGER	
NLUIS	0B	/LEVEL/		INTEGER	
SSABUSE	1B	/SAV		INTEGER	

```

--PROCEDURES--(LO-A)
--NAME--TYPE--ARGS--CLASS--
IRAY  INTEGER  2  FUNCTION

```

```

--STATEMENT LABELS--(LO-A)
--LABEL-ADDRESS--PROPERTIES--DEF

```

```

1  22B  110B  DO-TERM  33
10  INACTIVE  58  59
59B  110B

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--

```

```

FIND  5B  0

```

```

--STATISTICS--

```

```

PROGRAM-UNIT LENGTH  142B : 3B
CH LABELLED COMMON LENGTH  132B : 3B
CH STORAGE USED  60603B : 24960
COMPILE TIME  0.007 SECONDS

```

```

1  SUBROUTINE HELP
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

*****
THIS ROUTINE PROVIDES A BRIEF DESCRIPTION OF THE MAIN
OPTIONS. IT IS CALLED DIRECTLY IF OPTION #12EL22 IS
INPUT. IT IS AUTOMATICALLY INVOKED BY MADAM IF THE
USER INPUTS THREE SUCCESSIVE UNACCEPTABLE ENTRIES TO
MADAM.
*****
      CALLED BY: MADAM
*****
      VARIABLES USED: USER
*****
      MODIFIED: CH
*****
      CORTON/P/USER
      SAVE
      CHARACTER CH,USER*10
      PRINT*,THE FOLLOWING IS A BRIEF EXPLANATION OF THE OPTIONS. FOR
      PRINT*,MORE DETAIL, SEE THE USER'S MANUAL, 'USER',
      PRINT*,
      PRINT*,22X ATT 22X ATTRIBUTE LABEL ENTRY'
      PRINT*,
      PRINT*,22X COP 22X COPIES ONE NODE TO ANOTHER'
      PRINT*,
      PRINT*,22X DIS 22X DISPLAY OF ONE NODE'
      PRINT*,
      PRINT*,22X DON 22X DONE WITH WORK, SAVE ALL FILES'
      PRINT*,
      PRINT*,22X HEL 22X HEL 22X WILL REPEAT THIS INFORMATION'
      PRINT*,
      PRINT*,22X MOD 22X MODIFIES EXISTING TREE. NODE BY NODE'
      PRINT*,
      PRINT*,22X NEW 22X NEW TREE BUILDING DRIVER'
      PRINT*,
      PRINT*,22X NAM 22X SAME AS REV, BUT WITH WEIGHTS AND VALUES'
      PRINT*,
      PRINT*,          (PRESS ANY LETTER TO CONTINUE)'
      PRINT*,
      PRINT*,          9,
      PRINT*,          READ(2, '(A1)',IOCM
      PRINT*,          22X PRU 22X PRUNES THE TREE NODES'
      PRINT*,
      PRINT*,          22X REV 22X REVISED PRINT OF TREE'
      PRINT*,
      PRINT*,          22X SEL 22X SELECT TREE FILE (STORED DATA)'
      PRINT*,
      PRINT*,          22X SEN 22X CONDUCT SENSITIVITY ANALYSIS'
*****

```

SUBROUTINE HELP

```

58 PRINT, 'SP4 ADDS DOWNLINKS TO EXISTING NODES'
59 PRINT, 'STA PROVIDES TREE STATISTICS'
60 PRINT, 'SYS INPUT ALTERNATIVE SYSTEMS'
61 PRINT, 'TTL DATA FILE TITLE ENTRY'
62 PRINT, 'WAC LOADS WEIGHTS AND VALUES, DOES CALCULATIONS'
63 PRINT, ' (PRESS ANY LETTER TO CONTINUE)'
64 READ, '(A1)'
65 RETURN
66 END

```

```

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

```

```

CH      1B /SSASUSE/ SAV      CHAR21
USER     0B /?/              CHAR210

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--

```

```

HELP     SB      0

```

---STATISTICS---

```

PROGRAM-UNIT LENGTH      4218 . 273
CH LABELLED COMMON LENGTH 33 . 3
CH STORAGE USED          696008 . 24963
COMPILE TIME              0.109 SECONDS

```

```

1  SUBROUTINE INTRO
2  C
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C

```

THIS ROUTINE ESTABLISHES AN IDENTIFICATION OF THE USER
 FOR EMBELLISHING SEVERAL PROMPTS.

CALLED BY: MADAM

VARIABLES
 USED: (NONE)
 MODIFIED: USER

COMMON/P/USER
 SAVE
 CHARACTER*10 USER
 PRINT*, 'WHAT IS YOUR NAME, PLEASE?'
 READ(*, '(A10)') USER
 PRINT*, 'THANK YOU, 'USER''. WE WILL NOW BEGIN THE '
 PRINT*, 'DECISION ANALYSIS.'
 RETURN
 END

```

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

```

```

USER      6B  /P/      CHARACTER10

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--

```

```

INTRO      5B  0

```

```

--STATISTICS--

```

```

PROGRAM-UNIT LENGTH      62B  50
CR LABELLED COMMON LENGTH  2B  2
CR STORAGE USED          64600B  24550
COMPILE TIME              0.02B SECONDS

```



```

1 SUBROUTINE 'SET(I,J,IM)
2 C
3 C=====
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C

```

THIS ROUTINE IS USED TO SET THE VALUES OF THE PSEUDO-
 ARRAY, VARIABLE IRAY(I,J). SEE FUNCTION IRAY(I,J) FOR
 A DESCRIPTION OF THE VALUES.

CALLED BY: MASTER, MODIFY, PRUNE, SPAN

VARIABLES
 USED: I, IM, J, NTAPE
 MODIFIED: IRAY, ISIN, K

COMMON/CTRL/ISAUD, NTAPE
 COMMON/IRAY1/IRAY(20)
 COMMON/IRAY2/ISIN, NLOUD
 SAVE
 IF(I.NE.ISIN)READ(NTAPE,REC=1,ERR=99)(IRAY(K),K=1,80)
 IRAY(J)=IM
 WRITE(NTAPE,REC=1,ERR=99)(IRAY(K),K=1,80)
 ISIN=I
 RETURN
 PRINT*, 'I/O ERROR IN ISET'
 RETURN
 END

99

SUBROUTINE ISET

--ENTRY POINTS--(LO=A)
 --NAME--ADDRESS--ARGS--
 ISET 58 3

--STATISTICS--

PROGRAM-UNIT LENGTH 648 * 52
 CM LABELLED COMMON LENGTH 1268 * 86
 CM STORAGE USED 606008 * 24060
 COMPILE TIME 0.039 SECONDS

--VARIABLE MAP--(LO=A)		--BLOCK--		--PROPERTIES--		--TYPE--		--SIZE--	
NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS
I	1	DUMMY-ARG	2	DUMMY-ARG	2	INTEGER	13	IRAY	13
IM	3	DUMMY-ARG	13	IRAY1	13	INTEGER	13	ISAUD	13
IRAY	08	IRAY1	08	IRAY2	13	INTEGER	13	NTAPE	13
ISAUD	08	IRAY2	08	IRAY2	13	INTEGER	13	NTAPE	13
ISIN	08	IRAY2	08	IRAY2	13	INTEGER	13	NTAPE	13

--STATEMENT LABELS--(LO=A)
 --LABEL-ADDRESS--PROPERTIES--DEF

59 332 33


```

SUBROUTINE NEXT
C *****
C THIS ROUTINE IS USED TO PERFORM A NODE-BY-NODE DEPTH-
C FIRST SEARCH OF THE TREE. A NEW NODE IS ACCESSED EACH
C TIME THAT THIS ROUTINE IS CALLED.
C *****
C CALLED BY: COPYR,CHECK,DELSYS,GRAFIX,RDATT,SPAN,CALC,
C          LULODI,MJHREV
C *****
C VARIABLES USED: NLULS
C *****
C MODIFIED: ICONT,IDATA,IFADD,IFIND,INCRN,LEVEL,
C          LUL
C *****
COMMON/LEVEL/NLULS, INCRN(20), IFIND,NDIFF, IFADD,LUL, LEVEL(20,3)
COMMON/MEX/ICONT, IDATA, ITOTL
SAVE
ICONT=0
IDATA=0
IF ((LUL.GT.0).THEN
    IF ((LEVEL(LUL,2)/NE.1)).AND.(IRAY(LEVEL(LUL,1),2).GT.0)) THEN
        LEVEL(LUL,2)=1
        LEVE=(LUL+1,1)=IRAY(LEVEL(LUL,1),2)
        LEVEL(LUL+1,2)=0
        LUL=LUL+1
        IFIND=LEVEL(LUL,1)
        ICONT=1
        IFADD=2
        GO TO 99
    ENDIF
ELSE
    IF ((LUL.EQ.1).AND.(NLULS.NE.0.)GO TO 99
    IF ((IRAY(IFIND,3).GT.0)) THEN
        LEVEL(LUL,1)=IRAY(IFIND,3)
        LEVEL(LUL,2)=0
        IFIND=LEVEL(LUL,1)
        ICONT=1
        IFADD=1
        IFADD=3
        GO TO 99
    ENDIF
    LUL=LUL-1
    IF ((LUL.GT.0)) THEN
        IFIND=LEVEL(LUL,1)
        GO TO 1
    ENDIF
    IF ((IRAY(IFIND,2).LE.0)) IDATA=1
    IF ((LUL.GT.0).AND.(LUL.LT.21)) INCRN(LUL)=IRAY(IFIND,1)
END

```

SUBROUTINE NEXT

```

50
50
50
50
ENDIF
RETURN
END

```

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ICONT	0B	/NEX/		INTEGER	
IDATA	1B	/NEX/		INTEGER	
IFADD	67B	/LEVEL/		INTEGER	
IFIND	25B	/LEVEL/		INTEGER	
INRKN	1B	/LEVEL/		INTEGER	
ITOTL	2B	/NEX/		INTEGER	
LEVEL	31B	/LEVEL/		INTEGER	
LU	30B	/LEVEL/		INTEGER	
NDIFF	26B	/LEVEL/		INTEGER	
RLVLS	4B	/LEVEL/		INTEGER	

```

--PROCEDURES--(LO-A)
--NAME-----TYPE-----ARGS-----CLASS-----
IRAY  INTEGER  2  FUNCTION

```

```

--STATEMENT LABELS--(LO-A)
--LABEL-ADDRESS-----PROPERTIES-----DEF
1  54B  42
99 113B 56

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
NEXT  5B  0

```

---STATISTICS---

```

PROGRAM-UNIT LENGTH      1561 - 110
CR LABELLED COMMON LENGTH 1318 - 89
CM STORAGE USED          606000 - 24860
COMPILE TIME              0.003 SECONDS

```



```

1 SUBROUTINE OBJECT(I,OBJECTU)
2
3 *****
4
5 THIS ROUTINE ACCESSES THE OBJECTIVE ASSOCIATED WITH A
6 PARTICULAR MODE. OBJECT(I,OBJECTU) STORES THE OBJECTIVE
7 OF MODE I IN THE VARIABLE OBJECTU. THE OBJECTIVES ARE
8 SET VIA THE SUBROUTINE OSET(I,OBJECTU).
9
10 *****
11
12 CALLED BY: COPYR, GRAFIX, PRUNE, RDATT, SPON, GRUEC, RDV,
13          RDUT, DISPLA, MURREV, DETNOD, GRAPH, MEASUR,
14          MOUTSET
15
16 *****
17
18 VARIABLES
19 USED: I, NTAPE
20
21 MODIFIED: ISIN, J, OBJECTU
22
23 *****
24
25 COMMON /CTRL/ ISAUD, ITAPE
26 COMMON /RRAY1/ IRRAY(80)
27 COMMON /RRAY2/ ISIN, MLOUD
28
29 SAVE
30
31 CHARACTER*10 OBJECTU(16), IRRAY
32 IF(I.NE.ISIN)READ(UNIT=REC-1,ERR=99)(IRRAY(J),J=1,50)
33 DO 10 J=1,16
34   OBJECTU(J)=IRRAY(J+64)
35   ISIN=1
36   RETURN
37   PRINT*, 'I/O ERROR IN OBJECT'
38   RETURN
39   END
40
41 *****
42
43
44 *****
45
46
47 *****
48
49
50 *****
51
52 *****
53
54 *****
55
56 *****
57
58 *****
59
60 *****
61
62 *****
63
64 *****
65
66 *****
67
68 *****
69
70 *****
71
72 *****
73
74 *****
75
76 *****
77
78 *****
79
80 *****
81
82 *****
83
84 *****
85
86 *****
87
88 *****
89
90 *****
91
92 *****
93
94 *****
95
96 *****
97
98 *****
99
100 *****
101
102 *****
103
104 *****
105
106 *****
107
108 *****
109
110 *****
111
112 *****
113
114 *****
115
116 *****
117
118 *****
119
120 *****
121
122 *****
123
124 *****
125
126 *****
127
128 *****
129
130 *****
131
132 *****
133
134 *****
135
136 *****
137
138 *****
139
140 *****
141
142 *****
143
144 *****
145
146 *****
147
148 *****
149
150 *****
151
152 *****
153
154 *****
155
156 *****
157
158 *****
159
160 *****
161
162 *****
163
164 *****
165
166 *****
167
168 *****
169
170 *****
171
172 *****
173
174 *****
175
176 *****
177
178 *****
179
180 *****
181
182 *****
183
184 *****
185
186 *****
187
188 *****
189
190 *****
191
192 *****
193
194 *****
195
196 *****
197
198 *****
199
200 *****
201
202 *****
203
204 *****
205
206 *****
207
208 *****
209
210 *****
211
212 *****
213
214 *****
215
216 *****
217
218 *****
219
220 *****
221
222 *****
223
224 *****
225
226 *****
227
228 *****
229
230 *****
231
232 *****
233
234 *****
235
236 *****
237
238 *****
239
240 *****
241
242 *****
243
244 *****
245
246 *****
247
248 *****
249
250 *****
251
252 *****
253
254 *****
255
256 *****
257
258 *****
259
260 *****
261
262 *****
263
264 *****
265
266 *****
267
268 *****
269
270 *****
271
272 *****
273
274 *****
275
276 *****
277
278 *****
279
280 *****
281
282 *****
283
284 *****
285
286 *****
287
288 *****
289
290 *****
291
292 *****
293
294 *****
295
296 *****
297
298 *****
299
300 *****
301
302 *****
303
304 *****
305
306 *****
307
308 *****
309
310 *****
311
312 *****
313
314 *****
315
316 *****
317
318 *****
319
320 *****
321
322 *****
323
324 *****
325
326 *****
327
328 *****
329
330 *****
331
332 *****
333
334 *****
335
336 *****
337
338 *****
339
340 *****
341
342 *****
343
344 *****
345
346 *****
347
348 *****
349
350 *****
351
352 *****
353
354 *****
355
356 *****
357
358 *****
359
360 *****
361
362 *****
363
364 *****
365
366 *****
367
368 *****
369
370 *****
371
372 *****
373
374 *****
375
376 *****
377
378 *****
379
380 *****
381
382 *****
383
384 *****
385
386 *****
387
388 *****
389
390 *****
391
392 *****
393
394 *****
395
396 *****
397
398 *****
399
400 *****
401
402 *****
403
404 *****
405
406 *****
407
408 *****
409
410 *****
411
412 *****
413
414 *****
415
416 *****
417
418 *****
419
420 *****
421
422 *****
423
424 *****
425
426 *****
427
428 *****
429
430 *****
431
432 *****
433
434 *****
435
436 *****
437
438 *****
439
440 *****
441
442 *****
443
444 *****
445
446 *****
447
448 *****
449
450 *****
451
452 *****
453
454 *****
455
456 *****
457
458 *****
459
460 *****
461
462 *****
463
464 *****
465
466 *****
467
468 *****
469
470 *****
471
472 *****
473
474 *****
475
476 *****
477
478 *****
479
480 *****
481
482 *****
483
484 *****
485
486 *****
487
488 *****
489
490 *****
491
492 *****
493
494 *****
495
496 *****
497
498 *****
499
500 *****
501
502 *****
503
504 *****
505
506 *****
507
508 *****
509
510 *****
511
512 *****
513
514 *****
515
516 *****
517
518 *****
519
520 *****
521
522 *****
523
524 *****
525
526 *****
527
528 *****
529
530 *****
531
532 *****
533
534 *****
535
536 *****
537
538 *****
539
540 *****
541
542 *****
543
544 *****
545
546 *****
547
548 *****
549
550 *****
551
552 *****
553
554 *****
555
556 *****
557
558 *****
559
560 *****
561
562 *****
563
564 *****
565
566 *****
567
568 *****
569
570 *****
571
572 *****
573
574 *****
575
576 *****
577
578 *****
579
580 *****
581
582 *****
583
584 *****
585
586 *****
587
588 *****
589
590 *****
591
592 *****
593
594 *****
595
596 *****
597
598 *****
599
600 *****
601
602 *****
603
604 *****
605
606 *****
607
608 *****
609
610 *****
611
612 *****
613
614 *****
615
616 *****
617
618 *****
619
620 *****
621
622 *****
623
624 *****
625
626 *****
627
628 *****
629
630 *****
631
632 *****
633
634 *****
635
636 *****
637
638 *****
639
640 *****
641
642 *****
643
644 *****
645
646 *****
647
648 *****
649
650 *****
651
652 *****
653
654 *****
655
656 *****
657
658 *****
659
660 *****
661
662 *****
663
664 *****
665
666 *****
667
668 *****
669
670 *****
671
672 *****
673
674 *****
675
676 *****
677
678 *****
679
680 *****
681
682 *****
683
684 *****
685
686 *****
687
688 *****
689
690 *****
691
692 *****
693
694 *****
695
696 *****
697
698 *****
699
700 *****
701
702 *****
703
704 *****
705
706 *****
707
708 *****
709
710 *****
711
712 *****
713
714 *****
715
716 *****
717
718 *****
719
720 *****
721
722 *****
723
724 *****
725
726 *****
727
728 *****
729
730 *****
731
732 *****
733
734 *****
735
736 *****
737
738 *****
739
740 *****
741
742 *****
743
744 *****
745
746 *****
747
748 *****
749
750 *****
751
752 *****
753
754 *****
755
756 *****
757
758 *****
759
760 *****
761
762 *****
763
764 *****
765
766 *****
767
768 *****
769
770 *****
771
772 *****
773
774 *****
775
776 *****
777
778 *****
779
780 *****
781
782 *****
783
784 *****
785
786 *****
787
788 *****
789
790 *****
791
792 *****
793
794 *****
795
796 *****
797
798 *****
799
800 *****
801
802 *****
803
804 *****
805
806 *****
807
808 *****
809
810 *****
811
812 *****
813
814 *****
815
816 *****
817
818 *****
819
820 *****
821
822 *****
823
824 *****
825
826 *****
827
828 *****
829
830 *****
831
832 *****
833
834 *****
835
836 *****
837
838 *****
839
840 *****
841
842 *****
843
844 *****
845
846 *****
847
848 *****
849
850 *****
851
852 *****
853
854 *****
855
856 *****
857
858 *****
859
860 *****
861
862 *****
863
864 *****
865
866 *****
867
868 *****
869
870 *****
871
872 *****
873
874 *****
875
876 *****
877
878 *****
879
880 *****
881
882 *****
883
884 *****
885
886 *****
887
888 *****
889
890 *****
891
892 *****
893
894 *****
895
896 *****
897
898 *****
899
900 *****
901
902 *****
903
904 *****
905
906 *****
907
908 *****
909
910 *****
911
912 *****
913
914 *****
915
916 *****
917
918 *****
919
920 *****
921
922 *****
923
924 *****
925
926 *****
927
928 *****
929
930 *****
931
932 *****
933
934 *****
935
936 *****
937
938 *****
939
940 *****
941
942 *****
943
944 *****
945
946 *****
947
948 *****
949
950 *****
951
952 *****
953
954 *****
955
956 *****
957
958 *****
959
960 *****
961
962 *****
963
964 *****
965
966 *****
967
968 *****
969
970 *****
971
972 *****
973
974 *****
975
976 *****
977
978 *****
979
980 *****
981
982 *****
983
984 *****
985
986 *****
987
988 *****
989
990 *****
991
992 *****
993
994 *****
995
996 *****
997
998 *****
999
1000 *****

```

```

1 SUBROUTINE OSET(I,OBJECT)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1
```

```

10 2
11 3
12 4
13 5
14 6
15 7
16 8
17 9
18 10
19 11
20 12
21 13
22 14
23 15
24 16
25 17
26 18
27 19
28 20
29 21
30 22
31 23
32 24
33 25
34 26
35 27
36 28
37 29
38 30
39 31
40 32
41 33
42 34

```

```

FUNCTION PARAM(I,J)
C
C *****
C THIS FUNCTION SIMULATES A PSEUDO-ARRAY VARIABLE
C PARAM(I,J). PARAM(I,1) IS THE INTERCEPT (B0) OF THE
C LINEAR REGRESSION REPRESENTING THE INDIVIDUAL VALUE
C FUNCTION OF ATTRIBUTE # PARAM(I,2) IS THE SLOPE
C COEFFICIENT (B1). PARAM(I,3) IS THE SUM OF SQUARED
C ERROR. PARAM(I,4) IS THE INDEX DENOTING THE VALUES
C ARE SET BY THE SUBROUTINE PSET(I,J,U).
C
C CALLED BY: VALU,XLEVEL,PICTUR,VALUE
C
C VARIABLES
C USED: 1,MLOAD,NTAPE
C MODIFIED: ARRAY,ISIN,J,JJ,K
C *****
COMMON/CTRL/ISAUD,NTAPE
COMMON/PRAY/ARRAY(80)
COMMON/PRAY2/ISIN,NLOAD
SAVE
J=J
J=MLOAD+J-1
IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
PARAM=ARRAY(I)
ISIN=J
J=JJ
RETURN
PRINT*,I/O ERROR IN PARAM
RETPUH
END
90

```

```

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--
ARRAY 00 /PRAY/
I 1 /PRAY-ARG
ISAUD 48 /CTRL/
ISIN 80 /PRAY2/
J 12 /PRAY-ARG

```

```

--PROPERTIES--
REAL
INTEGER
INTEGER
INTEGER
INTEGER

```

```

--SIZE--
80

```

```

--FUNCTION MAP--(LO-A)
--NAME--ADDRESS--BLOCK--
PARAM 00 /PRAY/

```

```

--PROPERTIES--
REAL
INTEGER
INTEGER
INTEGER
INTEGER

```

```

--SIZE--
40

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
PARAM 60 2

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH 600 * 48
CR LABELLED COMMON LENGTH 1278 * 87
CR STORAGE USED 60000 * 24000
CR FILE TIME 9.937 SECONDS

```

```

1  SUBROUTINE PRENEX
2
3  C*****
4  C
5  C THIS ROUTINE INITIALIZES THE PROGRAM TO BEGIN A
6  C DEPTH-FIRST SEARCH, USING SUBROUTINE NEXT. FROM A
7  C DESIRED NODE. ANALOGOUS TO PRETOT.
8  C
9  C
10 C
11 C CALLED BY: COPYR.SPAN,VALOR;MUR,BETNOD,MODSET
12 C
13 C
14 C VARIABLES
15 C USED: NDIFF,MLULS,MNODES
16 C
17 C MODIFIED: ICONT,IDATA,IFIND,INNRH,ITOTL,LEVEL,
18 C LUL
19 C
20 C*****
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

SUBROUTINE PRENEX

---VARIABLE NAME---	---(LO-A)---	---BLOCK---	---PROPERTIES---	---TYPE---	---SIZE---
ICONT	68	/NEX/		INTEGER	58
IDATA	18	/NEX/		INTEGER	
IFADD	278	/LEVEL/		INTEGER	
IFIND	258	/LEVEL/		INTEGER	
IMMKN	18	/LEVEL/		INTEGER	
ITOTL	28	/NEX/		INTEGER	

---PROCEDURES---	---(LO-A)---	---ARGS---	---CLASS---
IRAY	INTEGER	2	FUNCTION
MODIM		0	SUBROUTINE

---ENTRY POINTS---	---(LO-A)---
PRENEX	58 0

---STATISTICS---
PROGRAM-UNIT LENGTH
CM LABELLED COMMON LENGTH
CM STORAGE USED
COMPILE TIME

```

SUBROUTINE PRETOT
C*****
C THIS ROUTINE INITIALIZES THE PROGRAM TO BEGIN A DEPTH-
C FIRST SEARCH, USING SUBROUTINE NEXT, FROM THE ROOT
C NODE. ANALOGOUS TO PRENEX.
C
C CALLED BY: DELEVS, RDATT, RDEVS, SPAN, CALC, WULOD, MURREV,
C          SENSTU
C
C VARIABLES USED: MNODES
C MODIFIED: ICONT, IDATA, IFIND, INNM, ITOTL, LEVEL,
C          LUL, NOIFF, NLULS
C*****
COMMON/C/MNODES, MDEEP
COMMON/LEVEL/NLULS, INNM(20), IFIND, NOIFF, IFADD, LUL, LEVEL(20,2)
COMMON/INEX/ICONT, IDATA, ITOTL
SAVE
ICONT=0
IDATA=0
LEVEL(1,1)=2
LEVEL(1,2)=0
LUL=1
NLULS=0
NOIFF=1
IFIND=LEVEL(1,1)
IF (IRAV(IFIND,2).LE.Q) IDATA=1
IF (INNMES.GT.1) ICONT=1
INNM(1)=IRAV(IFIND,1)
ITOTL=1
RETURN
END

```

NAME	ADDRESS	SIZE	TYPE	PROPERTIES	BLOCK	ADDRESS	SIZE	TYPE	PROPERTIES
ICONT	62	1	INTEGER			318	1	LEVEL	INTEGER
IDATA	12	1	INTEGER			320	1	LEVEL	INTEGER
IFADD	278	1	LEVEL			14	1	NOIFF	INTEGER
IFIND	280	1	LEVEL			268	1	LEVEL	INTEGER
INNM	18	20	INTEGER			48	1	NOIFF	INTEGER
ITOTL	25	1	INTEGER			60	1	LEVEL	INTEGER

--STATISTICS--
 PROGRAM-UNIT LENGTH 608 * 48
 CM LABELLED COMMON LENGTH 1333 * 91
 CM STORAGE USED 608088 * 24960
 COMPILE TIME 9.845 SECONDS

```

1  SUBROUTINE PSET(I,J,U)
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C

```

THIS ROUTINE SETS THE VALUES OF THE PSEUDO-ARRAY
 VARIABLE PARAM(I,J). PSET(X,J,U) SETS THE J-TH
 PARAMETER ASSOCIATED WITH X-TH PSEUDO-ARRAY. SEE
 PARAM(I,J) FOR AN EXPLANATION OF THE PARAMETERS.
 THE VALUES ARE ACCESSED VIA FUNCTION PARAM(I,J).

CALLED BY: REGRS

VARIABLES
 USED: NLOD,NTAPE,U

MODIFIED: ARRAY,ISIN,J,J,K

COMMON/CTRL/ISAUD,NTAPE
 COMMON/PRAY1/ARRAY(80)
 COMMON/PRAY2/ISIN,NLOD
 SAVE
 JJ=J
 J=NLOD+J-1
 IF(J.NE.ISIN)READ(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
 ARRAY(I)=U
 WRITE(NTAPE,REC=J,ERR=99)(ARRAY(K),K=1,80)
 ISIN=J
 J=J
 RETURN
 PRINT*,I/O ERROR IN PSET
 RETURN
 END

99

ENTRY POINTS--(LO-A)
 NAME ADDRESS ARGV

PSET 58 3

STATISTICS--
 PROGRAM-UNIT LENGTH 753 59
 CH LABELLED COMMON LENGTH 1278 87
 CH STORAGE USED 585508 24853
 COMPILE TIME 0.944 SECONDS

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ARRAY	08	1	ARRAY-ARG	REAL	80
ISAUD	08	2	CTRL	INTEGER	1
ISIN	08	3	ARRAY-ARG	INTEGER	1
J	08	4	ARRAY-ARG	INTEGER	1
SUBROUTINE PSET	74/74	0FT=0			
JJ	12	1	SSAUSE/SAU	INTEGER	1
K	28	2	SSAUSE/SAU	INTEGER	1
NLOD	12	3	CTRL	INTEGER	1
NTAPE	12	4	CTRL	INTEGER	1
U	3	5	PRAY-ARG	REAL	1
FTN 5.1-528					

STATEMENT LABELS--(LO-A)
 LABEL ADDRESS PROPERTIES BCF

99 428 32

```

1  FUNCTION SYSLEL(I)
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
```



```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE TLOAD
C
C*****
C THIS ROUTINE LOADS OR INITIALIZES A DATA FILE.
C
C CALLED BY: RADAN, INIT, TAPER
C
C VARIABLES
C   USED: NTAPE
C
C   MODIFIED: CH, I, IARRAY, ISIN, ISTOP, J, NATT, MLOAD,
C             MNODES, NSVS
C*****
C
C COMMON/C/MNODES, NDEEP
C COMMON/CTRL/ISAD, NTAPE
C COMMON/REAY1/IARRAY(88)
C COMMON/REAY2/ISIN, MLOAD
C COMMON/SYS1/NSVS
C SAVE
C CHARACTER CH
C MLOAD=565
C ISIN=0
C
C OPEN(NTAPE, ERR=99, ACCESS='DIRECT', RECL=100)
C PRINT*, 'OPENING FILE NUMBER ', NTAPE
C PRINT*, 'IS THIS DATA NEW (N) OR STORED (S)?'
C READ(1, '(A1)') CH
C IF ((CH.EQ.'N').AND.(CH.NE.'S')) THEN
C   PRINT*, 'PLEASE ENTER "N" OR "S".'
C   GO TO 1
C ENDIF
C IF (CH.EQ.'N') THEN
C   DO 10 I=1, 88
C     IARRAY(I)=0
C   ISTOP=MLOAD+10
C   DO 20 I=1, ISTOP
C     WRITE(NTAPE, REC=1, ERR=99) (IARRAY(J), J=1, 80)
C     MNODES=0
C   ELSE
C     I=MLOAD+6
C     READ(NTAPE, REC=1, ERR=99) MNODES
C     I=MLOAD+3
C     REAP(NTAPE, REC=1, ERR=99) NATT
C     I=MLOAD+8
C     READ(NTAPE, REC=1, ERR=99) NSVS
C   ENDIF
C   RETURN
C PRINT*, 'I/O ERROR IN TLOAD'
C
1
16
20
50

```

SUBROUTINE TLOAD

```

58      RETURN
59      END

```

```

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
CH      12 /SEASUSE/ SAV      INTEGER
I      28 /SEASUSE/ SAV      INTEGER
IARRAY  38 /ARRAY/          INTEGER
IGAND   48 /CNTRL/          INTEGER
ISIN    58 /ARRAY/          INTEGER
ISTOP   68 /SEASUSE/ SAV      INTEGER

```

```

--STATEMENT LABELS--(LO-A)
--LABEL--ADDRESS--PROPERTIES--DEF
1      158 INACTIVE DO-TERM 33
20     160 INACTIVE DO-TERM 43
58     1228 INACTIVE DO-TERM 57

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
TLOAD  518 0

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH      2478 - 167
CM LABELLED COMMON LENGTH 1358 - 93
CM STORAGE USED          685888 - 24988
COMPILE TIME              0.009 SECS

```

```

1  SUBROUTINE TSAVE
2
3  C
4  C *****
5  C THIS ROUTINE CONVERTS THE CURRENT DATA FILE TO A LOCAL
6  C FILE, STORING ALL INFORMATION.
7  C *****
8  C
9  C CALLED BY: MADAR, INITT, TAPER
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C

```

VARIABLES
 USED: NATT, NLOAD, NNODES, NSYS, NTAPE
 MODIFIED: I, ISAUD

COMMON/C/NNODES, NDEEP
 COMMON/CHTML/ISAUD, HTAPE
 COMMON/RRAY1/IRRAY(MB)
 COMMON/RRAY2/ISIN, FLOAD
 COMMON/SYS1/NSYS
 SAVE

I=NLOAD-4
 WRITE(TAPE,REC=1,ERR=99)NNODES
 I=NLOAD-3
 WRITE(TAPE,REC=1,ERR=99)NATT
 I=NLOAD-2
 WRITE(TAPE,REC=1,ERR=99)NSYS
 CLOSE(TAPE,ERR=99)
 ISAUD=1
 RETURN
 PRINT*, 'I/O ERROR IN TSAVE'
 RETURN
 END

99

SUBROUTINE TSAVE

---STATEMENT LABELS---(LO-A)		---LABEL-ADDRESS---		---PROPERTIES---		---DEF	
99	338						33
---ENTRY POINTS---(LO-A) ---NAME---ADDRESS---ARGS--- TSAVE 59 0							
---STATISTICS--- PROGRAM-UNIT LENGTH 739 59 CH LABELLED COMMON LENGTH 1328 99 CH STORAGE USED 606008 24960 COMPILE TIME 0.048 SECONDS							
---VARIABLE MAP---(LO-A)		---BLOCK---		---PROPERTIES---		---TYPE---	
NAME	ADDRESS	BLOCK	NAME	ADDRESS	SIZE	TYPE	SIZE
I	18	SSAVE/	NDEEP	18	18	INTEGER	18
ISAUD	43	RRAY1/	NLOAD	18	18	INTEGER	18
ISIN	68	CHTML/	NNODES	68	68	INTEGER	68
NATT	83	RRAY2/	NSYS	83	83	INTEGER	83
	98	SSAVE/	NTAPE	98	98	INTEGER	98

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
FUNCTION VALU(X,LABEL)
*****
THIS FUNCTION RETURNS THE VALUE OF A PARTICULAR
ATTRIBUTE LEVEL (X) FOR AN ATTRIBUTE (LABEL) USING
THE INDIVIDUAL VALUE FUNCTION.
*****
CALLED BY: RDU, COMPUT
*****
VARIABLES
  USED: LABEL, NATT, K
  MODIFIED: VALU
*****
FUNCTION VALU
*****
COMMON/ATTR/NATT
SAVE
CHARACTER*10 LABEL,ATT;
VALU=0.0
DO 10 I=1,NATT
  IF (LABEL.EQ.ATT(I)) THEN
    IF (PARAM(I,4).EQ.1.0) X=X*2
    IF (PARAM(I,4).EQ.-1.0) X=X*2.5
    IF (PARAM(I,4).EQ.2.0) X=EXP(X)
    IF (PARAM(I,4).EQ.-2.0) X=ALOG(X)
    VALU=PARAM(I,1)+PARAM(I,2)*X
  GO TO 20
ENDIF
CONTINUE
RETURN
END
10 99
*****
--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
1 LABEL 1B /SOURCE/ SAV  INTEGER
2 NATT 2B /ATTR/  DUMMY-ARG  INTEGER
3 VALU 3B /SOURCE/ SAV  REAL
4 X 1  DUMMY-ARG  REAL
*****
--PROCEDURES--(LO-A)
--NAME--TYPE--ARCS--CLASS--
ALOG REAL 1 INTRINSIC
ATT1 CHARACTER*10 1 FUNCTION
EXP REAL 1 INTRINSIC
PARAM REAL 2 FUNCTION
*****
--STATEMENT LABELS--(LO-A)
--LABEL--ADDRESS--PROPERTIES--DEF
10 INACTIVE DO-TERM 37
20 752 38
*****
--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARCS--
VALU 6B 2
*****
--STATISTICS--
PROGRAM-UNIT LENGTH 1438 90
CH LABELLED COMMON LENGTH 38 3
% STORAGE USED 66606 24063
% CPU TIME 0.054 SECONDS

```

```

18 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
FUNCTION URAY(I,J)
C*****
C THIS FUNCTION SIMULATES THE ARRAY VARIABLE URAY(I,J).
C URAY(I,J) IS THE VALUE OF SYSTEM J AT NODE I. THESE
C VALUES ARE SET BY USET(I,J,U).
C*****
C CALLED BY: DELSYS,CALC.REV,DISPLA,DSPLT,MU'PREU,COMPUT.
C*****
C VARIABLES USED: I,J,NTAPE
C MODIFIED: ARRAY,IPULL,ISIN,K
C*****
COMMON/CTRL/ISAUD,NTAPE
COMMON/ARRAY/ARRAY(80)
COMMON/ARRAY2/ISIN,NLOUD
SAVE
IPULL=7+J
IF(I.NE.ISIN)READ(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
URAY=ARRAY(IPULL)
ISIN=I
RETURN
PRINT*, 'I/O ERROR IN URAY'
RETURN
END
90
C*****
C
C*****
FUNCTION VRAY
C*****
C ENTRY POINTS--(LO,0)
C NAME--ADDRESS--ARGS--
C VRAY 58 2
C*****
C STATISTICS--
C PROGRAM-UNIT LENGTH 548 44
C CM LABELLED COMMON LENGTH 1278 87
C CM STORAGE USED 663008 24960
C CM CPU TIME 9.037 SECONDS
C*****
C VARIABLE MAP--(LO,0)
C NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
C ARRAY 08 /ARRAY/ REAL 80
C IPULL 1 DUMMY-ARG INTEGER 80
C ISAUD 18 /SEASUSE/ SAW
C ISIN 08 /CTRL/ INTEGER
C 08 /ARRAY2/ INTEGER
C*****
C STATEMENT LABELS--(LO,0)
C LABEL-ADDRESS--PROPERTIES--DEV
C 80 318
C 34

```

```

1 SUBROUTINE USET(I,J,U)
2 C
3 C*****
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C

```

THIS ROUTINE SETS THE VALUES OF THE PSEUDO-ARRAY
 VARIABLE URAY(I,J). USET(I,J,U) SETS THE VALUE OF
 SYSTEM J AT NODE I TO U. THESE VALUES ARE ACCESSSED
 VIA FUNCTION URAY(I,J).

CALLED BY: DELSYS,RDU,CALC

VARIABLES
 USED: I,J,NTAPE,U
 MODIFIED: ARRAY,IPUT,ISIN,K

COMMON/CTRL/ISAUD,NTAPE
 COMMON/RRAY1/ARRAY(80)
 COMMON/RRAY2/ISIN,ICLOUD
 SAVE
 IPUT=7+J
 IF(I.NE.ISIN)READ(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
 ARRAY(IPUT)=U
 WRITE(NTAPE,REC=1,ERR=99)(ARRAY(K),K=1,80)
 ISIN=1
 RETURN
 PRINT*, 'I/O ERROR IN USET'
 RETURN
 END

99

SUBROUTINE USET

--ENTRY POINTS--(LO-A)
 --NAME--ADDRESS--ARGS--
 USET 58 3

--STATISTICS--
 PROGRAM-UNIT LENGTH 678 * 55
 CM LABELLED COMMON LENGTH 1278 * 87
 CM STORAGE USED 805000 * 24860
 COMPILE TIME 0.041 SECONDS

NAME	ADDRESS	SIZE	TYPE	PROPERTIES	BLOCK	NAME	ADDRESS	SIZE	TYPE	PROPERTIES	BLOCK
ARRAY	08	80	REAL			DUMMY-ARG	2		INTEGER		
I	1		INTEGER			SSAUSE/SAU	2R		INTEGER		
IPUT	12		INTEGER			RRAY2/	1B		INTEGER		
ISAUD	08		INTEGER			CTRL/	1B		INTEGER		
ISIN	08		INTEGER			DUMMY-ARG	3		REAL		

--STATEMENT LABELS--(LO-A)
 --LABEL-ADDRESS--PROPERTIES--DEF

80 363 35

FUNCTION XLEVEL(X,LABEL)

 THIS FUNCTION RETURNS THE ATTRIBUTE LEVEL OF AN ATTRIBUTE
 (LABEL) FOR PARTICULAR VALUE (X) BASED ON THE INDIVIDUAL
 VALUE FUNCTION.

CALLER BY: RDV,SEXUAL

VARIABLES
 USED: LABEL,NATT,X
 MODIFIED: (NONE)

FUNCTION XLEVEL

```

COMMON/ATTR/NATT
SAVE
CHARACTER*10 LABEL,ATT;
XLEVEL=0.0
DO 10 I=1,NATT
  IF (LABEL.EQ.ATT(I)) THEN
    XLEVEL=(X-PARAM(I,1))/PARAM(I,2)
  IF (PARAM(I,4).EQ.1.0) XLEVEL=XLEVEL+5
  IF (PARAM(I,4).EQ.-1.0) XLEVEL=XLEVEL-2
  IF (PARAM(I,4).EQ.2.0) XLEVEL=XLEVEL+LOG(XLEVEL)
  IF (PARAM(I,4).EQ.-2.0) XLEVEL=XLEVEL*EXP(XLEVEL)
  GO TO 99
ENDIF
CONTINUE
RETURN
END
  
```

NAME	TYPE	ARGS	CLASS
ALOG	REAL	1	INTRINSIC
ATT1	CHAR*10	1	FUNCTION
EXP	GENERIC	1	INTRINSIC
PARAM	REAL	2	FUNCTION

STATEMENT LABELS--(LO-A)	PROPERTIES--DEF
10 INACTIVE	DO-TERM 37
99	763 38

ENTRY POINTS--(LO-A)
 NAME--ADDRESS--ARGS--

XLEVEL 68 2

STATISTICS--

PROGRAM-UNIT LENGTH 1458 - 101
 CM LABELLED COMMON LENGTH 38 - 3
 CM STORAGE USED 886992 - 24860
 COMPILE TIME 0.658 SECONDS

VARIABLE MAP--(LO-A)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
I	18	18	SSRUSC/ SAV	INTEGER	
LABEL	2	2	DUMMY-ARG	CHAR*10	
NATT	08	08	DUMMY-ARG	INTEGER	
X	1	1	DUMMY-ARG	REAL	
XLEVEL	08	08	SSRUSC/ SAV	REAL	

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55
OVERLAY(XFILE,1,0)
PROGRAM DUTRY
C
C*****
C THIS PROGRAM PROVIDES THE ENTRY POINT FOR
C OVERLAY(1,0).
C
C CALLED BY: (NONE)
C
C VARIABLES
C USED: CND
C MODIFIED: (NONE)
C
C*****
COMMON/ATTR/MATT
COMMON/CANODES,NDEEP
COMMON/CATRL/ISAUD,NTAPE
COMMON/LEVEL/ALULS,INSTR(25),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
COMMON/MEX/ICONT,IDATA,ITOTL
COMMON/ONE/CND
COMMON/P/USER
COMMON/RRAV1/ARRAY(80)
COMMON/RRAV2/ISIN,NL0UD
COMMON/SV31/MSYS
CHARACTER*3 CND
IF(CND.EQ.'ATT')THEN
CALL RDATT
ELSEIF(CND.EQ.'NGU')THEN
CALL TAPER
ELSEIF(CND.EQ.'SEL')THEN
CALL TAPER
ELSEIF(CND.EQ.'SPA')THEN
CALL SPAN
ELSEIF(CND.EQ.'TTL')THEN
CALL RTTL
ELSEIF(CND.EQ.'MOD')THEN
CALL MODIFY
ELSEIF(CND.EQ.'PRU')THEN
CALL PRUIN
ELSEIF(CND.EQ.'SYA')THEN
CALL STAY
ELSEIF(CND.EQ.'SYS')THEN
CALL RDSYSL
ENDIF
RETURN
END
RETURN IN MAIN PROGRAM -- ACTS AS END

```

```

PROGRAM DUMMY

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
LVL      08 /LEVEL/      INTEGER
NATT     42 /ATTR/       INTEGER
NDEEP    08 /LEVEL/      INTEGER
NDIFF    08 /LEVEL/      INTEGER
NLOUD    08 /LEVEL/      INTEGER
NLULS    08 /LEVEL/      INTEGER
NMADES    08 /C/         INTEGER
NPSYS    08 /SYS1/       INTEGER
NSTATE   08 /CNTRL/      INTEGER
USER     08 /P/          REAL

CHAR13    REAL          80
INTEGER    INTEGER      20
INTEGER    INTEGER      60
INTEGER    INTEGER
INTEGER    INTEGER
INTEGER    INTEGER
INTEGER    INTEGER
INTEGER    INTEGER

```

```

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
BDTTL     08 /LEVEL/      SUBROUTINE
SCAN      08 /C/         SUBROUTINE
STAT      08 /LEVEL/      SUBROUTINE
TAPER     08 /P/          SUBROUTINE

```

```

--PROCEDURES--(LO-A)
--NAME--TYPE--ARGS--CLASS--
MODIFY    0 SUBROUTINE
PRUNE     0 SUBROUTINE
RDATT     0 SUBROUTINE
RDSYSL    0 SUBROUTINE

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
DUMMY     0D 0

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH      08 - 0
CM LABELLED COMMON LENGTH 2623 - 178
CM STORAGE USED          606003 - 24960
COMPILE TIME             0.046 SECONDS
1 ANSI ERROR IN DUMMY

```

```

1  SUBROUTINE ADDSYS
2  C
3  C*****
4  C
5  C THIS ROUTINE ALLOWS THE EXTENSION OF AN EXISTING SET
6  C OF ALTERNATIVE SYSTEMS.
7  C
8  C CALLED BY: RDSYSL
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C

```

THIS ROUTINE ALLOWS THE EXTENSION OF AN EXISTING SET
 OF ALTERNATIVE SYSTEMS.
 CALLED BY: RDSYSL
 VARIABLES USED: (NONE)
 MODIFIED: ANSWER, ISAUD, NSYS

COMMON/SYS1/MSYS
 SAVE
 CHARACTER*10 SYSIDL,ANSWER
 PRINT*,
 IF (MSYS.EQ.50) THEN
 PRINT*, 'MAXIMUM NUMBER OF SYSTEMS EXCEEDED.'
 ELSE
 MSYS=MSYS+1
 WRITE(*, '(5(/,1X)')
 PRINT*, 'ADDING SYSTEM'
 READ(*, '(A10)') ANSWER
 CALL SYSSET(MSYS,ANSWER)
 PRINT*, 'USE *** LOC *** FOR ENTERING VALUES.'
 PRINT*, 'AND RECALCULATING TREE (IF NECESSARY).'
 ISAUD=0
 ENDIF
 RETURN
 END

SUBROUTINE ADDSYS
 PROCEDURES--(LO-A)
 NAME--TYPE--ARGS--CLASS--
 SYSSET 2 SUBROUTINE
 ENTRY POINTS--(LO-A)
 NAME--ADDRESS--ARGS--
 ADDSYS 5B 0
 STATISTICS--
 1312 - 89
 5B - 5 24869
 506002 - 24869
 0.644 SECONDS
 PROGRAM-LIMIT LENGTH
 CR LABELLED COMMON LENGTH
 CR STORAGE USED
 COMPILE TIME

VARIABLE MAP--(LO-A)
 NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
 ANSWER 2B /\$ANSWER/ SAV CHARACTER10
 ISAUD 3B /\$ISAUD/ SAV INTEGER
 MSYS 0B /\$MSYS/ SAV INTEGER
 SYSIDL 1B /\$SYSIDL/ UNUSED/SAV CHARACTER10

```

1 SUBROUTINE ASK(I,X1,X2,XMID)
2
3 C
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C

```

THIS ROUTINE ASKS THE QUESTIONS WHICH ELICIT THE
 DATA POINTS FROM WHICH THE ESTIMATED INDIVIDUAL
 VALUE FUNCTION MAY BE ESTABLISHED.

CALLED BY: VALUE

VARIABLES
 USED: X1,X2,XMAX,XMIN
 MODIFIED: XMID

SAVE
 CHARACTER ATT:10
 PRINT*, 'WHAT LEVEL OF 'ATT:10
 PRINT*, 'WOULD BE SUCH THAT YOU WOULD FEEL THE SAME'
 PRINT*, 'AMOUNT OF CHANGE IN SATISFACTION IN MOVING'
 PRINT*, 'FROM 'X1,' TO 'X2,' AS FROM THAT LEVEL TO 'X2,'?'
 PRINT*, ' 'X1,' TO 'X2,' AS FROM THAT LEVEL TO 'X2,'?'
 READ(*, '(F10.0)') XMID
 XMID=MIN(X1,X2)
 XMAX=MAX(X1,X2)
 IF ((XMID.LT.XMIN).OR.(XMID.GT.XMAX)) THEN
 PRINT*, 'XMID, ' IS OUTSIDE THE RANGE OF 'X1,' TO 'X2'
 GO TO 1
 ENDIF
 RETURN
 END

SUBROUTINE ASK

PROCEDURES--(LO-A)
 NAME--TYPE--AROS--CLASS--
 AROS1 REAL 7 INTRINSIC
 AROS2 REAL 7 INTRINSIC
 AROS3 CHAR10 1 FUNCTION

STATEMENT LABELS--(LO-A)
 LABEL-ADDRESS--PROPERTIES--DEF
 1 78 26

ENTRY POINTS--(LO-A)
 NAME--ADDRESS--AROS--
 ASK 58 4

STATISTICS--
 PROGRAM-UNIT LENGTH 1578 111
 CM LABELLED COMMON LENGTH 38 3
 CM STORAGE USED 66600 24880
 COMPILE TIME 0.053 SECONDS

VARIABLE MAP--(LO-A)
 NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
 I 1 DUMMY-ARG 18 /SUBROUTINE/ SAV REAL
 XMAX 28 /SUBROUTINE/ SAV REAL
 XMID 4 DUMMY-ARG 3 REAL

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE CHECK
C
C *****
C THIS ROUTINE IS USED TO CONDUCT THE BETWEEN NODE
C CHECK WHICH VALIDATES THE SET OF CHILDREN FOR THE
C CURRENT PARENT NODE.
C
C CALLED BY: SPAN
C
C VARIABLES
C   USED: CONTRL,NNODES,USER
C   MODIFIED: CH,ICHECK
C *****
C
COMMON/ONE/CHD
COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
COMMON/CHILD/CONTRL
COMMON/C/NNODES,NDEEP
COMMON/P/USER
COMMON/CHK/ICHECK
SAVE
CHARACTER*10 LABEL,USER,CH11,CRD13
INTEGER CONTRL
IF((NNODES.GT.2).AND.(CONTRL.GT.1))THEN
  ICHECK=0
  CALL GRAFIX
  PRINT,USER,' DO THE SUBOBJECTIVES ADDRESS ALL FACETS'
  PRINT,OF THE PARENT OBJECTIVE? (Y/N)'
  PRINT,
  READ(3, '(A1)')CH
  IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
    PRINT,'ONLY 'Y' AND 'N' ARE LEGITIMATE ENTRIES, 'USER
    GO TO 2
  ENDIF
  IF(CH.NE.'Y')THEN
    ICHECK=1
    CALL MODIFY
    GO TO 1
  ELSE
    PRINT,
    PRINT,' IS THERE ANY OVERLAP BETWEEN THE COVERAGES OF'
    PRINT,' THE SUBOBJECTIVES, 'USER,'? (Y/N)'
    PRINT,
    READ(2, '(A1)')CH
    IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT,'PLEASE ENTER 'Y' OR 'N', 'USER
      GO TO 3
    ENDIF
  ENDIF
END

```

SUBROUTINE CHECK

```

ENDIF
IF (C.H.NE.'N').THEN
  ICHECK=1
  CALL PRUKE
  GO TO 1
ELSE
  PRINT(2,'')
  PRINT(3,'USER',AREAD(1))
  PRINT(3,'MEANINGFUL',AREAD(2))
  READ(3, '(A1)')CH
  IF (C.H.NE.'Y').AND
    PRINT(3,'ONLY 'Y'. OK
  GO TO 4
ENDIF
IF (C.H.NE.'Y').THEN
  ICHECK=1
  CALL MODIFY
  GO TO 1
ELSE
  PRINT(3,'')
  PRINT(3,'COULD ANY
  PRINT(3,'SIGNIFICANT
  PRINT(3,'')
  READ(3, '(A1)')CH
  IF (C.H.NE.'Y').AND
    PRINT(3,'PLEASE ENTER
  GO TO 5
ENDIF
IF (C.H.NE.'N').THEN
  ICHECK=1
  CALL PRUKE
  GO TO 1
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
ICHECK=2
CALL STAT
CALL CALL NEXT
RETURN
END

```

```

4
ENDIF
IF(CH.NE.'N')THEN
  ICHECK=1
  CALL PRUNE
  GO TO 1
ELSE
  PRINT*,',',
  PRINT*,USER,',', ARE ALL THE SUBOBJECTIVES OPERATIONALLY ,
  PRINT*,MEANINGFUL TO YOU? (Y/N)',
  PRINT*,',',
  READ*,'(A1)',CH
  IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
    PRINT*,ONLY 'Y' OR 'N' IS ALLOWED, ',USER
    GO TO 4
  ENDIF
  IF(CH.NE.'Y')THEN
    ICHECK=1
    CALL MODIFY
    GO TO 1
  ELSE
    PRINT*,',',
    PRINT*,COULD ANY OF THE SUBOBJECTIVES BE IGNORED WITHOUT',
    PRINT*,SIGNIFICANTLY IMPACTING YOUR PREFERENCES, ',USER,'? (Y/N)',
    PRINT*,',',
    READ*,'(A1)',CH
    IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
      PRINT*,PLEASE ENTER 'Y' OR 'N', ',USER
      GO TO 5
    ENDIF
    IF(CH.NE.'N')THEN
      ICHECK=1
      CALL PRUNE
      GO TO 1
    ENDIF
    ENDIF
    ENDIF
    ENDIF
    ICHECK=0
    CALL STAT
    CALL NEXT
    RETURN
  END

```

```

5

```

```
-----PROCEDURES--(LO=9)-----CLASS-----
```

NAME	TYPE	ARGS
GRAFIX		@
MODIFY		@
NEXT		@
PRIME		@
STAT		@
SUBROUTINE		
SUBROUTINE		
SUBROUTINE		
SUBROUTINE		
SUBROUTINE		

```
-----STATEMENT LABELS--(LO=A)
-LABEL-ADDRESS-----PROPERTIES-----DEF
```

1	78	33
2	208	36
3	598	50
4	1118	64
5	1158	78

```

--ENTRY POINTS--(LO=4)
-NAME---ADDRESS--BSCS---

```

0 55 2034

--5J11512315--

PROGRAM-UNIT LENGTH 4403 • 282
CM LABELLED COMMON LENGTH 1208 • 54
CM STORAGE USED 60600 • 24960
COPILE TIME 0.157 SECONDS

-----TYPE-----313

INTEGER
INTEGER
INTEGER
INTEGER
CHAR(10)

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
...

2B	SSA/VE/ SAU	CHARL
07	ONE	CHICK
08	CHILD	CHICK
09	CHICK	CHICK
27B	LEVEL	CHICK
25B	LEVEL	CHICK
1B	LEVEL	CHICK
1B	SSA/VE/	CHICK
1B	LEVEL/SAU	CHICK

NAME	ADDRESS	-----BLOCK-----
LEVEL	318	/LEVEL/
LUL	342	/C/
NDDEP	1	/C/
NDIFF	288	/LEVEL/
NUIS	88	/LEVEL/
NUISL	88	/C/
NUDES	88	/C/
USER	88	/P/

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE DELSYS
C*****
C THIS ROUTINE IS USED TO DELETE AN ALTERNATIVE SYSTEM
C FROM AN EXISTING SET OF ALTERNATIVES.
C
C CALLED BY: RDSYSL
C
C VARIABLES
C   USED: IFIND,NNODES,MSYS
C   MODIFIED: ANSWER,I,ISTR,J,L,MSYS,TSYS
C*****
COMMON/C,NNODES,NDEEP
COMMON/SYS1/MSYS
SAVE
CHARACTER*10 SYSLBL,TSYS(80),ANSWER,ISTR
WRITE(2, '(5(/,1X)')
PRINT*, 'CURRENT SYSTEMS...'
DO 10 I=1,MSYS
  TSYS(I)=SYSLBL(I)
  PRINT*,SYSLBL(I)
CONTINUE
PRINT*, 'ENTER THE SYSTEM TO BE DELETED.'
PRINT*, '      ?'
READ(3, '(A10)')ANSWER
ANSWER=ANSWER//
ISTR=
DO 20 I=1,MSYS
  IF(TSYS(I).EQ.ANSWER)THEN
    ISTR=ANSWER
    L=I
  ENDIF
CONTINUE
IF(ISTR.EQ.'')THEN
  PRINT*, 'SYSTEM ',ANSWER,' NOT FOUND'
ELSE
  DO 30 J=L,79
    CALL SYSSET(I,TSYS(J+1))
    CALL PRSTOT
    DO 40 I=1,NNODES
      DO 40 J=L,79
        CALL USET(IFIND,J,URAY(IFIND,J+1))
        CALL NEXT
        CONTINUE
        MSYS=MSYS-1
      ENDIF
    ENDIF
  ENDIF

```

SUBROUTINE DELSYS

```
58      RETURN
59      END
```

```
--VARIABLE MAP--(LO*4)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
ANSWER 121B /SSABUSE/ SAV CHAR10
I       123B /SSABUSE/ SAV INTEGER
IFIND   126B /SSABUSE/ SAV INTEGER
ISTR    128B /SSABUSE/ SAV CHAR10
J       125B /SSABUSE/ SAV INTEGER
```

```
--PROCEDURES--(LO*4)
--NAME--TYPE--ARGS--CLASS--
NEXT     0 SUBROUTINE
PRETOT   0 SUBROUTINE
SYSLBL   1 FUNCTION
```

```
--STATEMENT LABELS--(LO*4)
--LABEL--ADDRESS--PROPERTIES--DEF
16 INACTIVE DO-TERM 32
28 INACTIVE DO-TERM 44
38 INACTIVE DO-TERM 49
40 INACTIVE DO-TERM 55
```

```
--ENTRY POINTS--(LO*4)
--NAME--ADDRESS--ARGS--
DELSYS 5B 0
```

```
--STATISTICS--
PROGRAM-UNIT LENGTH 26GB - 214
CN LABELLED COMMON LENGTH 132B - 56
CM STORAGE USED 68600B - 24564
COMPILE TIME 0.007 SECONDS
```

```

1  SUBROUTINE GRAFIX
2
3  C *****
4  C *****
5  C *****
6  C *****
7  C *****
8  C *****
9  C *****
10 C *****
11 C *****
12 C *****
13 C *****
14 C *****
15 C *****
16 C *****
17 C *****
18 C *****
19 C *****
20 C *****
21 C *****
22 C *****
23 C *****
24 C *****
25 C *****
26 C *****
27 C *****
28 C *****
29 C *****
30 C *****
31 C *****
32 C *****
33 C *****
34 C *****
35 C *****
36 C *****
37 C *****
38 C *****
39 C *****
40 C *****
41 C *****
42 C *****
43 C *****
44 C *****
45 C *****
46 C *****
47 C *****
48 C *****
49 C *****
50 C *****
51 C *****
52 C *****
53 C *****
54 C *****
55 C *****
56 C *****
57 C *****

```

THIS ROUTINE DRAWS A WIRING DIAGRAM OF THE CURRENT
PARENT NODE AND ITS SPAN OF CHILDREN.

CALLED BY: CHECK

VARIABLES USED: (NONE)

MODIFIED: CH, CONTRL, I, II, ICOUNT, INDEX, INMRN,
J,K, L, LEVEL, LUL, NCHILD, OBJECTU, TINNRN,
TLUL, TOP, TSTORE

COMMON/LEVEL/NULS, INMRN(20), IFIND, NDIFF, IFADD, LUL, LEVEL(20,3)
COMMON/CHILD/CONTRL
SAVE

CHARACTER CH, OBJECTU(16), F1(10), F1(17), F2(29), F3(31), F4(20), LK(2), TOP(16),
INTEGER CONTRL, TSTORE(20,3), TLUL, TINNRN(20)
DATA F1, F2, /('IX, IHG, 4A10, 1HG)', '(IX, I21, IHG, T30, 1HG, 4A10, 1HG)'/,
DATA F3, /('IX, T21, 1HG, 5C8, 8C8, 4A10, 1HG)'/,
DATA F4, /('IX, I21, IHG, T30, 4A2')/,
DO 5 J=1,3
DO 5 I=1,20
TSTORE(I,J)=LEVEL(I,J)
TLUL=LUL
NCHILD=CONTRL
ICOUNT=0
WRITE(X,'(5(/,1X)')'
ICOUNT=ICOUNT+1
IF(ICOUNT.EQ.1) THEN
CALL OBJECTU(LEVEL(LUL,I),OBJECTU)
DO 6 I=1,15
TOP(I)=OBJECTU(I)
DO 8 I=1,20
TINNRN(I)=INMRN(I)
ELSE
DO 7 I=1,15
OBJECTU(I)=TOP(I)
ENDIF
WRITE(X,'(20(IX,13))')(TIMR,.,I),I=1,LUL
PRINT*,L
II=?
DO 10 I=1,4
WRITE(X,FMT=F1)(OBJECTU(J),J=1,II+3)
II=II+4
END

SUBROUTINE GRAFIX

```

58      CONTINUE
59      PRINT*,L
60      IF (CONTRL.GE.3) THEN
61        INDEX=3
62      ELSE
63        INDEF=CONTRL
64      ENDIF
65      DO 30 K=1,INDEX
66        CALL OBJECT(LEVEL(LUL,1),OBJECTU)
67        WRITE(2,(1X,T21,1M0,T30,20(13,1X)),)((INNRN(I),I=1,LUL)
68        WRITE(2,FMT=F4)L
69        I1=1
70        DO 20 I=1,4
71          IF (I.NE.3) THEN
72            WRITE(2,FMT=F2)((OBJECTU(J),J=1,I,I=3)
73          ELSE
74            WRITE(2,FMT=F3)((OBJECTU(J),J=1,I,I=3)
75          ENDIF
76          I1=I+4
77        CONTINUE
78        WRITE(2,FMT=F4)L
79      CONTINUE
80      CONTRL=CONTRL-3
81      IF (CONTRL.GT.0) THEN
82        PRINT*, '(PRESS ANY LETTER TO SEE REMAINING SUBOBJECTIVES)'
83        PRINT*, ' '
84        READ(2, '(A1)') CH
85        GO TO 1
86      ENDIF
87      DO 40 J=1,3
88        DO 40 I=1,10
89          LEVEL(I,J)=TSTORE(I,J)
90        DO 50 I=1,20
91          INNRN(I)=TINNRN(I)
92        LUL=TLUL
93        CONTRL=CONTRL+1
94      RETURN
95      END
96

```

---VARIABLE NAME---	---ADDRESS---	---BLOCK---	---PROPERTIES---	---TYPE---	---SIZE---
CH	1B	/SSAUSE/	SAU	INTEGER	
CONTRL	0B	/CHILD/	SAU	INTEGER	
F1	23B	/SSAUSE/	SAU	CHARACTER	60
F2	24B	/SSAUSE/	SAU	INTEGER	
F3	27B	/SSAUSE/	SAU	INTEGER	
F4	33B	/SSAUSE/	SAU	INTEGER	
I	284B	/SSAUSE/	SAU	INTEGER	
ICOUNT	206B	/SSAUSE/	SAU	INTEGER	16
IFADD	27D	/LEVEL/	SAU	CHARACTER	20
IFIND	25A	/LEVEL/	SAU	INTEGER	16
I1	207B	/SSAUSE/	SAU	INTEGER	
INDEX	210B	/SSAUSE/	SAU	CHARACTER	16
INNRN	1B	/LEVEL/	SAU	INTEGER	32
J	207B	/SSAUSE/	SAU	INTEGER	
K	212	/SSAUSE/	SAU	CHARACTER	
L	10A	/SSAUSE/	SAU	INTEGER	
LEVEL	31A	/LEVEL/	SAU	INTEGER	
LUL	30A	/LEVEL/	SAU	INTEGER	
ACHILD	205B	/SSAUSE/	SAU	INTEGER	
NOIFF	26B	/LEVEL/	SAU	INTEGER	
NULS	0B	/LEVEL/	SAU	INTEGER	
OBJECTU	2A	/SSAUSE/	SAU	CHARACTER	16
TINNRN	157D	/SSAUSE/	SAU	INTEGER	20
TLUL	156B	/SSAUSE/	SAU	INTEGER	16
TOP	42B	/SSAUSE/	SAU	CHARACTER	16
TSTORE	02B	/SSAUSE/	SAU	INTEGER	32

SUBROUTINE GRAFIX

```

--PROCEDURE--(LO-A)
--NAME--TYPE-----ARGO-----CLAS-----
NEXT          9      SUBROUTINE
OBJECT        2      SUBROUTINE

```

```

--STATEMENT LABELS--(LO-A)
--LABEL-ADDRESS--PROPERTY-ES-----DEF
1 458      DO-TERM      40
5 INACTIVE DO-TERM      36
6 INACTIVE DO-TERM      45
7 INACTIVE DO-TERM      50

```

```

--LABEL-ADDRESS--PROPERTY-ES-----DEF
8 INACTIVE DO-TERM      47
10 INACTIVE DO-TERM      58
29 INACTIVE DO-TERM      78

```

```

--LABEL-ADDRESS--PROPERTY-ES-----DEF
30 INACTIVE DO-TERM      80
40 INACTIVE DO-TERM      90
50 INACTIVE DO-TERM      92

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGO-----
GRAFIX      58      0

```

---STATISTICS---

```

PROGRAM-UNIT LENGTH      5552 * 265
CH LABELLED COMMON LENGTH 3408 * 224
CH STORAGE USED          60608 * 24960
COMPILE TIME              0.204 SECONDS

```

```

1  SUBROUTINE INITT
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C

```

THIS ROUTINE INITIALIZES THE CONSTRUCTION OF A NEW
 DATA STRUCTURE.

CALLED BY: TAPER

VARIABLES
 USED: (NONE)

MODIFIED: (NONE)

CALL RDSYSL
 CALL RDTTL
 CALL MASTER
 CALL SPAN
 CALL TSAUE
 CALL "LOAD
 RET "

END

PROCEDURES--(LO-A)	NAME	TYPE	ARGS	CLASS	NAME	TYPE	ARGS	CLASS
MASTER			0	SUBROUTINE	SPAN		0	SUBROUTINE
RDSYSL			0	SUBROUTINE	TLOAD		0	SUBROUTINE
RDTTL			0	SUBROUTINE	TSAUE		0	SUBROUTINE

ENTRY POINTS--(LO-A)
 NAME ADDRESS ARGS

INITT 5B 0

STATISTICS--

PROGRAM-UNIT LENGTH
 OR STORAGE USED
 COMPILE TIME

13B 27
 66500B 24064
 0.028 SECONDS

```

1  SUBROUTINE MASTER
2  C
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C

```

THIS ROUTINE SETS UP A DUMMY ROOT NODE FROM
 WHICH THE TREE IS TO BE BUILT.

CALLED BY: INIT

VARIABLES
 USED: (NONE)

MODIFIED: NNODES

COMMON/C/NNODES,NDEEP

SAVE
 CALL ISET(1,'MASTER')
 CALL ISET(1,1,0)
 CALL ISET(1,3,-1)
 CALL ISET(1,4,0)
 NNODES=1
 RETURN
 END

SUBROUTINE MASTER

VARIABLE MAP--(LO-A)
 NAME ADDRESS BLOCK PROPERTIES TYPE SIZE STATISTICS
 NDEEP 18 /C/ INTEGER
 NNODES 08 /C/ INTEGER

PROGRAM-UNIT LENGTH
 CM LABELLED COMMON LENGTH
 CM STORAGE USED
 CM COMPILER USED
 CM COMPILE TIME

543 - 44
 33 - 3
 606008 - 24060
 0.028 SECONDS

PROCEDURES--(LO-A)
 NAME TYPE ARGS CLASS
 ISET 3 SUBROUTINE
 ISET 2 SUBROUTINE

ENTRY POINTS--(LO-A)
 NAME ADDRESS ARGS
 MASTER 5D 0

SUBROUTINE MODIFY

```

50 CALL OSET(1,IND,ANSWER)
51 ELSE
52 IF(1,CHECK.EQ.1)CONTROL=CONTROL+1
53 JSTART=NLULS-NDIFF+1
54 DO 10 J=JSTART,NLULS
55   NNODES=NNODES+1
56   CALL ISET(1,IND,1,FADD,NNODES)
57   CALL OSET(NNODES,1,1,INRM(1))
58   CALL ISET(NNODES,2,1)
59   CALL ISET(NNODES,3,1)
60   CALL ISET(NNODES,4,1,IND)
61   CALL ASET(NNODES,1,0,0)
62   CALL ASET(NNODES,2,0,0)
63   IFIND=NNODES
64   IFIND=2
65   CONTINUE
66   CALL OSET(NNODES,ANSWER)
67   ENDDIF
68   ENDDIF
69   IF(IQUIT.NE.1)GO TO 1
70   NLULS=NLULS
71   LUL=TLUL
72   DO 21 I=1,20
73     LEVEL(I,J)=TSTORE(I,J)
74     INRM(1)=INRM(1)
75     RETURN
76     ND
77   20
78   21
79   22
80   23
81   24
82   25
83   26
84   27
85   28
86   29
87   30

```

SUBROUTINE MODIFY

```

--STATEMENT LABELS--(LO-A)
--LABEL-ADDRESS--PROPERTIES--DEF
1 478 INACTIVE DO-TERM 43
5 INACTIVE DO-TERM 39
6 INACTIVE DO-TERM 40

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARCS--
MODIFY 53 0

```

--STATISTICS--

```

PROGRAM-LIMIT LENGTH 4118 * 255
CM LABELLED COMMON LENGTH 3018 * 193
CM STORAGE USED 606302 * 24950
COMPILE TIME 0.144 SECONDS

```

--VARIABLE MAP--(LO-A)		--PROPERTIES--		--BLOCK--		--ADDRESS--		--TYPE--		--SIZE--	
NAME	ADDRESS	BLOCK	PROPERTIES	NAME	ADDRESS	BLOCK	PROPERTIES	NAME	ADDRESS	BLOCK	PROPERTIES
ANSWER	18	/SSAUSE/	SAU	CHAR=10	16	LEVEL	/LEVEL/	INTEGER	318	/LEVEL/	60
CONTROL	08	/CHILD/	SAU	INTEGER		LUL	/LEVEL/	INTEGER	308	/LEVEL/	
I	148	/SSAUSE/	SAU	INTEGER		NDEEP	/C/	INTEGER	10	/C/	
ICHECK	08	/CHCX/		INTEGER		NDIFF	/LEVEL/	INTEGER	260	/LEVEL/	
IFADD	278	/LEVEL/		INTEGER		NLULS	/C/	INTEGER	08	/C/	
IFIND	258	/LEVEL/		INTEGER		NNODES	/SSAUSE/	INTEGER	28	/SSAUSE/	20
INRM	18	/LEVEL/		INTEGER		TINRM	/SSAUSE/	INTEGER	1158	/SSAUSE/	
IQUIT	1438	/SSAUSE/	SAU	INTEGER	20	TLUL	/SSAUSE/	INTEGER	1158	/SSAUSE/	
ISAUD	1468	/SSAUSE/	SAU	INTEGER		TLULS	/SSAUSE/	INTEGER	1428	/SSAUSE/	
J	1452	/SSAUSE/	SAU	INTEGER		TSTORE	/SSAUSE/	REAL	218	/SSAUSE/	60
JSTART	1478	/SSAUSE/	SAU	INTEGER							

```

--PROCEDURES--(LO-A)
--NAME--TYPE--ARCS--CLASS--

```

```

ASCT 3 SUBROUTINE
IGET 3 SUBROUTINE
MODIN 0 SUBROUTINE
OSET 2 SUBROUTINE

```

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
SUBROUTINE NEW
C*****
C THIS ROUTINE ALLOWS THE GENERATION OF AN INITIAL
C SET OF ALTERNATIVE SYSTEMS.
C
C CALLED BY: RDSYSL
C
C VARIABLES
C   USED: (NONE)
C
C MODIFIED: ANSWER,ISAUD,NSYS
C*****
COMMON/VSYS/NSYS
SAVE
CHARACTER*10 SYSLBL,ANSWER
WRITE(X,'(5(/),1X)')
PRINTX,' '
NSYS=1
PRINTX,' '
PRINTX,'ENTER...SYSTEM ' NSYS, ' LABEL'
PRINTX,'(10 LETTERS OR LFSS)'
PRINTX,' '
READ(X,'(A10)')ANSWER
CALL SYSSET(NSYS,ANSWER)
IF((SYSLBL(NSYS).NE.'DONE').AND.(SYSLBL(NSYS).NE.' '))THEN
  NSYS=NSYS+1
  GO TO 1
ENDIF
NSYS=NSYS-1
IF(NSYS.LT.1)THEN
  PRINTX,'YOU MUST ENTER AT LEAST ONE SYSTEM'
  GO TO 11
ENDIF
ISAUD=0
RETURN
END

```

SUBROUTINE NEW

```

--PROCEDURES--(LO=A)
--NAME--TYPE--ARGO--CLASS--
SYSLBL  CHAR10  1  FUNCTION
SYSSET  1  118  27
--STATEMENT LABELS--(LO=A)
--LABEL-ADDRESS--PROPERTIES--DEF
1 152 29
11 118 27
--ENTRY POINTS--(LO=A)
--NAME--ADDRESS--ARGS--
NEW 5B 0
--STATISTICS--
PROGRAM-UNIT LENGTH 1638 115
CM LABELLED COMMON LENGTH 48 4
CM LABELLED COMMON LENGTH 866000 24960
CM STORAGE USED 0.055 SECONDS
COMPILE TIME

```

```

--VARIABLE MAP--(LO=A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
ANSWER 1B /SBAUSE/ SAU CHARACTER
ISAUD 2B /SBAUSE/ SAU INTEGER
NSYS 3B /SBS1/ INTEGER

```

```

1  SUBROUTINE PICTUR(INDEX)
2
3  C*****
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

THIS ROUTINE GENERATES A PLOT OF THE INDIVIDUAL VALUE
 FUNCTION OF A PARTICULAR ATTRIBUTE (INDEX).

CALLED BY: VALUE

VARIABLES USED: INDEX

MODIFIED: FACLUL, I, ISUB, J, L, L2, LINE, TLUL, XLUL, XUAL

C*****

COMMON/ATTR/MATT

SAVE

CHARACTER LINE(0:40), F1(20), F2(14), L3(41), L2(41), ATT(1810)

DATA F1/'(IX,F7.2,SX,2H-41A1)'/

DATA F2/'(14X,1H+,41A1)'/

DATA L2/'0.0'.25.5.75.1.0'/

FACLUL=1.0

DO 10 I=1,20

DO 20 J=0,40

LINE(J),

XLUL=(ATT(INDEX,2)-ATT(INDEX,1))*FACLUL+ATT(INDEX,1)

IF (PARAM(INDEX,4).EQ.0) TLUL=XLUL

IF (PARAM(INDEX,4).EQ.1) TLUL=XLUL**2

IF (PARAM(INDEX,4).EQ.-1) TLUL=XLUL**5

IF (PARAM(INDEX,4).EQ.-2) TLUL=ALOG(XLUL)

IF (PARAM(INDEX,4).EQ.-3) TLUL=EXP(XLUL)

XUAL=PARAM(INDEX,1)*PARAM(INDEX,2)*TLUL

IF (XUAL.GT.1.0) XUAL=1.0

IF (XUAL.LT.0.0) XUAL=0.0

ISUB=INT(XUAL*40.0)

LINE(ISUB)=J

IF ((I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.

(I.EQ.20)) THEN

WRITE(3,FMT=F1) XLUL, (LINE(J), JJ=0,40)

ELSE

WRITE(3,FMT=F2)((LINE(JJ), JJ=0,40))

ENDIF

FACLUL=FACLUL*.05

CONTINUE

WRITE(3,'(15X,A41)') L

WRITE(3,'(15X,A41)') L2

PRINT*,

PRINT*,

SUBROUTINE PICTURE

```

58 PRINT, 'VALUE FUNCTION FOR ', ATT1(INDEX)
59 RETURN
60 END
61

```

--VARIABLE MAP--(LO-A)		--PROPERTIES--		--TYPE--		--SIZE--		--NAME--		--ADDRESS--		--BLOCK--		--PROPERTIES--		--TYPE--		--SIZE--	
NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE		
FACUL	25B	/SASUSE/	SAU	REAL		LINE	13B	/SASUSE/	SAU	CHAR141	41								
F1	6B	/SASUSE/	SAU	CHAR122		L2	1B	/SASUSE/	SAU	CHAR131									
F2	11B	/SASUSE/	SAU	CHAR114		NATT	26B	/SASUSE/	SAU	CHAR141									
I	26B	/SASUSE/	SAU	INTEGER		TLUL	5B	/ATTR/		INTEGER									
INDEX	1	DUMMY-ARG		INTEGER		XLUL	31B	/SASUSE/	SAU	REAL									
ISUB	33B	/SASUSE/	SAU	INTEGER		XUL	30B	/SASUSE/	SAU	REAL									
J	27B	/SASUSE/	SAU	INTEGER			32B	/SASUSE/	SAU	REAL									
JJ	34B	/SASUSE/	SAU	INTEGER						REAL									

--PROCEDURES--(LO-A)		--CLASS--		--NAME--		--TYPE--		--CLASS--	
NAME	ADDRESS	TYPE	CLASS	NAME	ADDRESS	TYPE	CLASS	NAME	ADDRESS
ALOG	REAL	1	INTRINSIC	EXP		GENERIC			
ATT	REAL	2	FUNCTION	INT		GENERIC			
ATT1	CHAR10	1	FUNCTION	PARAM		REAL			

--STATEMENT LABELS--(LO-A)

--LABEL-ADDRESS--(LO-A)

NAME	ADDRESS	TYPE	CLASS
10	INACTIVE	DO-TERM	53
20	INACTIVE	DO-TERM	34

--ENTRY POINTS--(LO-A)

--NAME--ADDRESS--ARGS--

NAME	ADDRESS	ARGS
PICTUR	5B	1

--STATISTICS--

PROGRAM-UNIT LENGTH	365B	245
OR LABELLED COMMON LENGTH <td>36B</td> <td>30</td>	36B	30
OR STORAGE USED <td>6000B</td> <td>24568</td>	6000B	24568
CORFILE TIME <td>0.136</td> <td>SECONDS</td>	0.136	SECONDS

```

1 SUBROUTINE PPI
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

C*****
C THIS ROUTINE CONDUCTS A TEST OF PAIRWISE PREFERENTIAL
C INDEPENDENCE BETWEEN THE ATTRIBUTES.
C*****
C CALLED BY: RDATT
C*****
C VARIABLES
C   USED:  NATT
C   MODIFIED:  ATTR1,ATTR2,BAND,CH,D1,DEL1,DEL2,I,
C             INDERR,ITOL,J,K,MAX1,MAX2,MIN1,MIN2,
C             TEMP,TEMP1,TEMP2,TOLER,TOTHER
C*****
C COMMON/ATTR/NATT
C COMMON/P/USER
C SAVE
C CHARACTER*10 ATTR1,ATTR2,TOTHER,ATT1,USER,CH*1
C REAL BAND(2,2),MAX1,MIN1,MAX2,MIN2
C INDERR=0
C PRINT*, 'AT WHAT TOLERANCE DO YOU WANT TO CHECK YOUR'
C PRINT*, 'RESPONSES, 'USER,' (PLUS OR MINUS X PERCENT)?'
C PRINT*, 'X=?'
C READ(3,'(I2)',ITOL)
C TOLER=FLOAT(ITOL)/100.0
C PRINT*, 'WE ARE WORKING AT PLUS OR MINUS ',ITOL,' PERCENT.'
C DO 10 I=1,NATT-1
C   ATTR1=ATT1(I)
C   MAX1=AMAX1(ATT1(I),ATT1(I,2))
C   MIN1=AMIN1(ATT1(I),ATT1(I,2))
C   DEL1=(MAX1-MIN1)/10.0
C   DO 20 J=I+1,NATT
C     ATTR2=ATT1(J)
C     MAX2=AMAX1(ATT1(J),ATT1(J,2))
C     MIN2=AMIN1(ATT1(J),ATT1(J,2))
C     DEL2=(MAX2-MIN2)/10.0
C     PRINT*, 'SUPPOSE THAT THE FOLLOWING'
C     PRINT*, 'ATTRIBUTES ARE AT THESE LEVELS:'
C     DO 30 K=1,NATT
C       IF((K.EQ.1).OR.(K.EQ.J))GO TO 30
C       TOTHER=ATT1(K)
C       TEMP=(ATT(K,2)-ATT(I,1))*25*ATT(K,1)
C       PRINT*,TOTHER,'*',TEMP
C       CONTINUE
C     PRINT*, 'THAT IS AT THE 25 PERCENT LEVEL'
C   10 CONTINUE
C 20 CONTINUE
C 30 CONTINUE
C*****

```

SUBROUTINE PPI

```

1
58 PRINTZ, '
59 TEMP1=MAX1-S.03DEL1
60 D1=TEMP1*100
61 TEMPE=MAX2-S.03DEL2
62 PRINTZ, 'YOU SUPPOSE THAT YOU HAVE THE INITIAL CONDITIONS:
63 PRINTZ, ATTR1, '.,', TEMP1, ' AND ', ATTR2, '.,', TEMP2
64
65 TEMP2=TEMP2-DEL2*3.0
66 PRINTZ, 'IMAGINE THAT ', ATTR2, ' IS CHANGED TO ', TEMP2
67 PRINTZ, 'WHAT LEVEL OF ', ATTR1, ' WOULD KEEP YOU AS SATISFIED'
68 PRINTZ, 'AS YOU WERE UNDER THE INITIAL CONDITIONS?'
69 PRINTZ, '(REMEMBER THAT ALL OTHER ATTRIBUTES ARE AT'
70 PRINTZ, ' THE 25 PERCENT LEVEL)'
71
72 READ(2, '(F10.0),', TEMP1
73 IF((TEMP1.LT.MIN1).OR.(TEMP1.GT.MAX1))THEN
74 PRINTZ, 'THE INPUT LEVEL OF ', TEMP1
75 PRINTZ, ' IS OUTSIDE THE GIVEN RANGE OF ', ATTR1(1,1), ' TO ', ATTR1(1,2)
76 GO TO 1
77 ENDIF
78
79 BAND(1,1)=TEMP1*D1
80 BAND(1,2)=TEMP1-D1
81 TEMP1=MAX1-S.03DEL1
82 TEMP2=MAX2-S.03DEL2
83
84 PRINTZ, 'SUPPOSE THAT YOU ARE STARTING AT'
85 PRINTZ, ATTR1, '.,', TEMP1, ' AND ', ATTR2, '.,', TEMP2
86
87 TEMP2=TEMP2+3.03DEL2
88 PRINTZ, 'IMAGINE THAT ', TEMP2, ' IN ', ATTR2, ' IS ACHIEVED.'
89 PRINTZ, 'TO WHAT LEVEL WOULD YOU CHANGE ', ATTR1, ' IN ORDER TO'
90 PRINTZ, 'REMAIN AS SATISFIED AS YOU WERE INITIALLY?'
91 PRINTZ, '(REMEMBER THAT ALL OTHER ATTRIBUTES ARE AT THE '
92 PRINTZ, ' 25 PERCENT LEVEL)'
93
94 READ(2, '(F10.0),', TEMP1
95 IF((TEMP1.LT.MIN1).OR.(TEMP1.GT.MAX1))THEN
96 PRINTZ, 'THE INPUT LEVEL OF ', TEMP1
97 PRINTZ, ' IS OUTSIDE THE GIVEN RANGE OF ', ATTR1(1,1), ' TO ', ATTR1(1,2)
98 GO TO 2
99 ENDIF
100
101 BAND(2,1)=TEMP1*D1
102 BAND(2,2)=TEMP1-D1
103
104 PRINTZ, 'USER, ' SUPPOSE NOW THAT THE FOLLOWING ATTRIBUTES'
105 PRINTZ, 'ARE SHIFTED TO THESE LEVELS:'
106 DO 40 K=1,NATT
107 IF((K.EQ.1).OR.(K.EQ.2))GO TO 40
108 TOTHE=ATTR1(K)
109 TEMP=(ATTR1(2)-ATTR1(K))*75+ATTR1(K,1)
110 PRINTZ, 'TOTHE, '.,', TEMP
111 CONTINUE
112 PRINTZ, 'THAT IS AT THE 75 PERCENT LEVEL'
113
114 TEMP1=MAX1-S.03DEL1
115 TEMP2=MAX2-S.03DEL2
116 PRINTZ, 'SUPPOSE THAT YOU HAVE'

```

SUBROUTINE PPI

```

115 PRINTZ,ATTR1,'.',TEMP1,' AND ',ATTR2,'.',TEMP2
116 PRINTZ
117 TEMP2=TEMP2-3.0DELE
118 PRINTZ,'IMAGINE THAT THE LEVEL OF ',ATTR2
119 PRINTZ,' IS CHANGED TO ',TEMP2,'
120 PRINTZ,'WOULD THE LEVEL OF ',ATTR1,' NEEDED TO REMAIN'
121 PRINTZ,'AS SATISFIED AS AT THE INITIAL CONDITIONS'
122 PRINTZ,'LIE BETWEEN ',BAND(1,1),' AND ',BAND(1,2)
123 PRINTZ,'(Y/N) ?'
124 READ(1,(A1))CH
125 IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
126 PRINTZ,'ONLY 'Y' OR 'N' IS ALLOWED, ',USER
127 GO TO 3
128 ENDIF
129 IF((CH.EQ.'N'))INDERR=1
130 PRINTZ
131 TEMP1=MAX1-5.0DELE
132 TEMP2=MAX2-5.0DELE
133 PRINTZ,'SUPPOSE THAT YOU HAVE THE INITIAL CONDITIONS'
134 PRINTZ,ATTR1,'.',TEMP1,' AND ',ATTR2,'.',TEMP2
135 PRINTZ
136 TEMP2=TEMP2+3.0DELE
137 PRINTZ,'IMAGINE THAT YOU MUST ACCEPT'
138 PRINTZ,'A LEVEL OF ',TEMP2,' IN ',ATTR2,'
139 PRINTZ,'WOULD THE LEVEL OF ',ATTR1
140 PRINTZ,'THAT YOU WOULD HAVE TO MOVE TO (IN ORDER TO BE AS'
141 PRINTZ,'SATISFIED AS UNDER THE INITIAL CONDITIONS) LIE'
142 PRINTZ,'BETWEEN ',BAND(2,1),' AND ',BAND(2,2)
143 PRINTZ,'(Y/N) ?'
144 READ(1,(A1))CH
145 IF((CH.NE.'Y').AND.(CH.NE.'N'))THEN
146 PRINTZ,USER,', YOU MUST ENTER 'Y' OR 'N'
147 GO TO 4
148 ENDIF
149 IF((CH.EQ.'N'))INDERR=1
150 IF((INDERR.EQ.0))THEN
151 PRINTZ,'THERE ARE NO INDEPENDENCE PROBLEMS'
152 PRINTZ,'WITH THE ATTRIBUTES TESTED SO FAR.'
153 PRINTZ,'DO YOU WISH TO ASSUME PRI FOR THE'
154 PRINTZ,'REMAINING ATTRIBUTES? (Y/N)'
155 PRINTZ
156 READ(1,(A1))CH
157 IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
158 PRINTZ,'PLEASE ENTER 'Y' OR 'N'
159 GO TO 5
160 ENDIF
161 IF((CH.EQ.'N'))THEN
162 PRINTZ,'EVER IF YOU DO NOT WISH TO ASSUME'
163 PRINTZ,'PRI AMONG THE REMAINING ATTRIBUTES'
164 PRINTZ,'DO YOU WANT TO STOP PPI TESTING? (Y/N)'
165 PRINTZ
166 READ(1,(A1))CH
167 IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
168 PRINTZ,'PLEASE ENTER 'Y' OR 'N'
169 GO TO 6
170 ENDIF
171

```

SUBROUTINE PPI

```

172 IF(CH.EQ.'Y')GO TO 7
173 ELSE
174 PRINT,'THERE ARE INDEPENDENCE PROBLEMS'
175 PRINT,'AMONG THE ATTRIBUTES ALREADY TESTED.'
176 PRINT,'DO YOU WANT TO STOP PPI TESTING? (Y/N)'
177 GO TO 5
178 PRINT,'?'
179 READ(2,'(A1)')CH
180 IF(CH.NE.'N').AND.(CH.NE.'Y'))THEN
181 PRINT,'PLEASE ENTER 'Y' OR 'N'.'
182 GO TO 2
183 ENDIF
184 IF(CH.EQ.'Y')GO TO 7
185 ENDIF
186 CONTINUE
187 IF(LINDERR.EQ.0)THEN
188 CALL VALUE
189 ELSE
190 PRINT,'THERE ARE INDEPENDENCE PROBLEMS'
191 PRINT,'AMONG THE ATTRIBUTES (PPI DOES'
192 PRINT,'NOT HOLD). DO YOU WISH TO'
193 PRINT,'CONTINUE THE ANALYSIS WITH AN'
194 PRINT,'ADDITIVE VALUE FUNCTION? (Y/N)'
195 PRINT,'?'
196 READ(2,'(A1)')CH
197 IF(CH.EQ.'Y')CALL VALUE
198 ENDIF
199 RETURN
200 END

```

--VARIABLE MAP--(LO-A)										--PROPERTIES--										--BLOCK--										--ADDRESS--										--NAME--										--TYPE--										--SIZE--																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																							
--NAME--										--ADDRESS--										--BLOCK--										--NAME--										--TYPE--										--CLASS--										--NAME--										--TYPE--										--CLASS--										--NAME--										--TYPE--										--CLASS--																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																					
ATTR1										1B										/SABUSE/										SAU										11B										/SABUSE/										SAU										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL										REAL									

SUBROUTINE PPI

```

--STATEMENT LABELS--(LO-A)
-LABEL-ADDRESS-----PROPERTIES-----DEF
  1 1673 58
  2 2543 82
  3 4128 111
  4 4728 130

-LABEL-ADDRESS-----PROPERTIES-----DEF
  10 INACTIVE DO-TERM 188
  20 INACTIVE DO-TERM 185
  30 160B DO-TERM 86
  40 483B DO-TERM 109

```

```

--ENTRY POINTS--(LO-A)
-NAME--ADDRESS--ARGS--
PPI 58 0

```

--STATISTICS--

```

PROGRAM-UNIT LENGTH 20053 * 1029
CM LABELLED COMMON LENGTH 333 * 27
CM LABELLED COMMON LENGTH 646838 * 27638
CM STORAGE USED 0.497 SECONDS
COMPILE TIME

```


SUBROUTINE PRUNE

```

58      GO TO 999
59      ENDIF
60      PRINT*
61      IFROM=2
62      PRINT* 'SELECT PRUNING OPTION:'
63      PRINT* 'MODE=DOWN D(OWN ONLY SELECT DOWN E(XIT)'
64      PRINT* '
65      READ(8,'(A1)')CH
66      IF(CH.EQ.'N')AND.(CH.NE.'D').AND.(CH.NE.'S').AND.
67      1 (CH.NE.'E')THEN
68      PRINT* 'YOU MUST SELECT 'N','D','S', OR 'E'
69      GO TO 3
70      ENDIF
71      IF(CH.EQ.'E')GO TO 999
72      IF(CH.EQ.'N')THEN
73      IF(ICHECK.EQ.1)CONTROL=CONTROL-1
74      LVL=LVL-1
75      IFFROM=IRAY(IFIND,4)
76      IF(IRAY(IFROM,3).NE.IFIND)THEN
77      CALL ISET(IFROM,2,IRAY(IFIND,3))
78      IF(IRAY(IFROM,2).GT.0)CALL ISET(IRAY(IFROM,2),4,IFROM)
79      GO TO 400
80      ELSE
81      CALL ISET(IFROM,3,IRAY(IFIND,3))
82      IF(IRAY(IFROM,3).GT.0)CALL ISET(IRAY(IFROM,3),4,IFROM)
83      GO TO 400
84      ENDIF
85      ELSEIF(CH.EQ.'D')THEN
86      CALL ISET(IFIND,2,-1)
87      GO TO 400
88      ELSEIF(CH.EQ.'S')THEN
89      IFIND=IRAY(IFIND,IFROM)
90      IF(IFIND.LE.0)GO TO 400
91      CALL OBJECT(IFIND,OBJECTV)
92      II=1
93      DO 10 J=1,4
94      WRITE(8,'(1X,4A10)')(OBJECTV(K),K=II,II+3)
95      II=II+4
96      CONTINUE
97      PRINT*
98      PRINT* 'UNICH IS CURRENT DESCENDANT NUMBER ',IRAY(IFIND,1)
99      PRINT* 'DO YOU WHICH TO ELIMINATE THIS DESCENDANT? (Y/N)'
100      PRINT* '
101      READ(8,'(A1)')CH
102      IF(CH.NE.'Y').AND.(CH.NE.'N')THEN
103      PRINT* 'ONLY 'Y' OR 'N' IS ALLOWED'
104      GO TO 5
105      ENDIF
106      IF(CH.EQ.'N')THEN
107      IFFROM=3
108      GO TO 4
109      ELSE
110      CALL ISET(IRAY(IFIND,4),IFROM,IRAY(IFIND,3))
111      IF(IRAY(IFIND,3).LE.0)GO TO 400
112      CALL ISET(IRAY(IFIND,3),4,IRAY(IFIND,4))
113      IFFROM=3
114

```

```

115 GO TO 4
116 ENDF
117 ENDF
118 IF(ND-LEVEL(LVL,1)
119 IDATA=0
120 IF(IPAY(FIND,2),LE.0)IDATA=0
121 GO TO 1
122 ENDF
123 PRINT,' '
124 LUL5=TLUL5
125 LUL=TLUL
126 DO 21 I=1,N3
127 DO 20 J=1,3
128 LEVEL(I,J)=TSTORE(I,J)
129 INNR(I)=TINNR(I)
130 RETURN
131 END

```

ENTRY POINTS--(LO-8)	NAME--ADDRESS--ARG--	PRIME	SP	0

--STATISTICS--

	PROGRAM--UNIT LENGTH	CM LABELLED COMMON LENGTH
	CM STORAGE USED	CM STORAGE USED
	COMPILE TIME	

734B - 176
305B - 197
60600B - 21550
6.352 SECND5

CURRILE	NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
	J	1452	/SSASUSE/	SAU	INTEGER	60
	K	1501	/SSASUSE/	SAU	INTEGER	
	L	312	/LEVEL/		INTEGER	
	LUL	303	/LEVEL/		INTEGER	
	LUL	268	/LEVEL/		INTEGER	
	NDIFF	02	/LEVEL/		INTEGER	
	HLVLS	12	/CNTL/		INTEGER	
	HTAPE	1243	/SSASUSE/	SAU	CHAR110	16
	NOCTU	762	/SSASUSE/	SAU	INTEGER	28
	TINNAM	758	/SSASUSE/	SAU	INTEGER	
	TLUL	1222	/SSASUSE/	SAU	INTEGER	60
	TLVLS	12	/SSASUSE/	SAU	REAL	
	TSTORE	12				

PROCEDURES--(LO=A)		ARGS	CLASS
NAME	TYPE		
IRAY	INTEGER	2	FUNCTION
ISCT		3	SUBROUTINE
MODIN		6	SUBROUTINE
		2	SUBROUTINE

OBJECT	STATEMENT LABELS--(LO*G)	DEF	-LABEL-ADDRESS-----	PROPERTIES-----	DEF	-LABEL-ADDRESS-----	PROPERTIES-----	DEF
1	78	26	5	2448	93	20	INACTIVE	128
2	522	47	10	INACTIVE	97	21	INACTIVE	129
3	1078	61	11	INACTIVE	41	400	DO-TERM	118
4	2348	61	12	INACTIVE	42	993	DO-TERM	123
		99						

```

18 SUBROUTINE RDATT
19 C
20 C*****
21 C THIS ROUTINE ELICITS THE ATTRIBUTE SET WHICH IS
22 C ASSOCIATED WITH THE OBJECTIVE HIERARCHY.
23 C
24 C CALLED BY: DUKTY,SPAR
25 C
26 C VARIABLES
27 C USED: ICONT, IDATA, LEVEL, LVL
28 C
29 C MODIFIED: ATTRIB, CH, II, ICONT, J, K, NATT, ORJCTU,
30 C YATT
31 C*****
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C
58 C
59 C
60 C
61 C
62 C
63 C
64 C
65 C
66 C
67 C
68 C
69 C
70 C
71 C
72 C
73 C
74 C
75 C
76 C
77 C
78 C
79 C
80 C
81 C
82 C
83 C
84 C
85 C
86 C
87 C
88 C
89 C
90 C
91 C
92 C
93 C
94 C
95 C
96 C
97 C
98 C
99 C
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
455 C
456 C
457 C
458 C
459 C
460 C
461 C
462 C
463 C
464 C
465 C
466 C
467 C
468 C
469 C
470 C
471 C
472 C
473 C
474 C
475 C
476 C
477 C
478 C
479 C
480 C
481 C
482 C
483 C
484 C
485 C
486 C
487 C
488 C
489 C
490 C
491 C
492 C
493 C
494 C
495 C
496 C
497 C
498 C
499 C
500 C
501 C
502 C
503 C
504 C
505 C
506 C
507 C
508 C
509 C
510 C
511 C
512 C
513 C
514 C
515 C
516 C
517 C
518 C
519 C
520 C
521 C
522 C
523 C
524 C
525 C
526 C
527 C
528 C
529 C
530 C
531 C
532 C
533 C
534 C
535 C
536 C
537 C
538 C
539 C
540 C
541 C
542 C
543 C
544 C
545 C
546 C
547 C
548 C
549 C
550 C
551 C
552 C
553 C
554 C
555 C
556 C
557 C
558 C
559 C
560 C
561 C
562 C
563 C
564 C
565 C
566 C
567 C
568 C
569 C
570 C
571 C
572 C
573 C
574 C
575 C
576 C
577 C
578 C
579 C
580 C
581 C
582 C
583 C
584 C
585 C
586 C
587 C
588 C
589 C
590 C
591 C
592 C
593 C
594 C
595 C
596 C
597 C
598 C
599 C
600 C
601 C
602 C
603 C
604 C
605 C
606 C
607 C
608 C
609 C
610 C
611 C
612 C
613 C
614 C
615 C
616 C
617 C
618 C
619 C
620 C
621 C
622 C
623 C
624 C
625 C
626 C
627 C
628 C
629 C
630 C
631 C
632 C
633 C
634 C
635 C
636 C
637 C
638 C
639 C
640 C
641 C
642 C
643 C
644 C
645 C
646 C
647 C
648 C
649 C
650 C
651 C
652 C
653 C
654 C
655 C
656 C
657 C
658 C
659 C
660 C
661 C
662 C
663 C
664 C
665 C
666 C
667 C
668 C
669 C
670 C
671 C
672 C
673 C
674 C
675 C
676 C
677 C
678 C
679 C
680 C
681 C
682 C
683 C
684 C
685 C
686 C
687 C
688 C
689 C
690 C
691 C
692 C
693 C
694 C
695 C
696 C
697 C
698 C
699 C
700 C
701 C
702 C
703 C
704 C
705 C
706 C
707 C
708 C
709 C
710 C
711 C
712 C
713 C
714 C
715 C
716 C
717 C
718 C
719 C
720 C
721 C
722 C
723 C
724 C
725 C
726 C
727 C
728 C
729 C
730 C
731 C
732 C
733 C
734 C
735 C
736 C
737 C
738 C
739 C
740 C
741 C
742 C
743 C
744 C
745 C
746 C
747 C
748 C
749 C
750 C
751 C
752 C
753 C
754 C
755 C
756 C
757 C
758 C
759 C
760 C
761 C
762 C
763 C
764 C
765 C
766 C
767 C
768 C
769 C
770 C
771 C
772 C
773 C
774 C
775 C
776 C
777 C
778 C
779 C
780 C
781 C
782 C
783 C
784 C
785 C
786 C
787 C
788 C
789 C
790 C
791 C
792 C
793 C
794 C
795 C
796 C
797 C
798 C
799 C
800 C
801 C
802 C
803 C
804 C
805 C
806 C
807 C
808 C
809 C
810 C
811 C
812 C
813 C
814 C
815 C
816 C
817 C
818 C
819 C
820 C
821 C
822 C
823 C
824 C
825 C
826 C
827 C
828 C
829 C
830 C
831 C
832 C
833 C
834 C
835 C
836 C
837 C
838 C
839 C
840 C
841 C
842 C
843 C
844 C
845 C
846 C
847 C
848 C
849 C
850 C
851 C
852 C
853 C
854 C
855 C
856 C
857 C
858 C
859 C
860 C
861 C
862 C
863 C
864 C
865 C
866 C
867 C
868 C
869 C
870 C
871 C
872 C
873 C
874 C
875 C
876 C
877 C
878 C
879 C
880 C
881 C
882 C
883 C
884 C
885 C
886 C
887 C
888 C
889 C
890 C
891 C
892 C
893 C
894 C
895 C
896 C
897 C
898 C
899 C
900 C
901 C
902 C
903 C
904 C
905 C
906 C
907 C
908 C
909 C
910 C
911 C
912 C
913 C
914 C
915 C
916 C
917 C
918 C
919 C
920 C
921 C
922 C
923 C
924 C
925 C
926 C
927 C
928 C
929 C
930 C
931 C
932 C
933 C
934 C
935 C
936 C
937 C
938 C
939 C
940 C
941 C
942 C
943 C
944 C
945 C
946 C
947 C
948 C
949 C
950 C
951 C
952 C
953 C
954 C
955 C
956 C
957 C
958 C
959 C
960 C
961 C
962 C
963 C
964 C
965 C
966 C
967 C
968 C
969 C
970 C
971 C
972 C
973 C
974 C
975 C
976 C
977 C
978 C
979 C
980 C
981 C
982 C
983 C
984 C
985 C
986 C
987 C
988 C
989 C
990 C
991 C
992 C
993 C
994 C
995 C
996 C
997 C
998 C
999 C
1000 C

```

SUBROUTINE RDATA

```

58      IF (CH.EQ.'N') GO TO 2
59      PRINT*, 'COULD THE ATTRIBUTE ATTRIB
60      PRINT*, 'BE CHANGED SO AS TO IMPROVE'
61      PRINT*, 'COMMUNICATING WHAT IS IMPLIED'
62      PRINT*, 'IN THE OBJECTIVE? (Y/N)'
63      PRINT*, '(1)'
64      READ(1, '(1)') CH
65      IF (CH.NE.'Y').AND.(CH.NE.'N')) THEN
66      PRINT*, 'PLEASE ENTER 'Y' OR 'N'
67      GO TO 4
68      ENDIF
69      IF (CH.EQ.'Y') GO TO 2
70      CALL ATTRSET(NATT,ATTRIB)
71      ENDIF
72      CALL NEXT
73      GO TO 1
74      ENDIF
75      INATT=NATT/4
76      DO 10 J=1,INATT
77      I=1
78      DO 11 K=J*4-3,J*4
79      TATT(I)=ATT(K)
80      I=I+1
81      I=I+1
82      WRITE(2, '(4(IX,A10))')(TATT(K),K=1,4)
83      IF (INATT*4.LT.NATT) THEN
84      I=1
85      DO 12 K=INATT*4+1,NATT
86      TATT(I)=ATT(K)
87      I=I+1
88      I=I+1
89      WRITE(2, '(4(IX,A10))')(TATT(K),K=1,I-1)
90      ENDIF
91      PRINT*, '
92      PRINT*, 'THE ABOVE IS THE CURRENT SET OF ATTRIBUTES, 'USER,'
93      PRINT*, 'IF YOU SEE ANY WHICH ARE REDUNDANT, OR'
94      PRINT*, 'WHICH HAVE A DIRECT IMPACT ON ONE ANOTHER'
95      PRINT*, '(E.G. HEIGHT AND THRUST),
96      PRINT*, 'YOU SHOULD REFORM THE ATTRIBUTE SET TO'
97      PRINT*, 'REMOVE THESE PROBLEMS.'
98      PRINT*, 'DO YOU WISH TO REFORM THE ATTRIBUTE SET, 'USER,'
99      PRINT*, '(Y/N)'
100      READ(2, '(A1)') CH
101      IF (CH.NE.'Y').AND.(CH.NE.'N')) THEN
102      PRINT*, 'ONLY 'Y' OR 'N' IS ALLOWED. 'USER
103      GO TO 5
104      ENDIF
105      IF (CH.EQ.'Y') GO TO 2
106      PRINT*, 'DO YOU WISH TO BYPASS INDEPENDENCE TESTING?'
107      READ(2, '(A1)') CH
108      IF (CH.NE.'Y') THEN
109      CALL MPI
110      ELSE
111      CALL UALGE
112      ENDIF
113      RETURN
114

```

SUBROUTINE RDATT

END

116

--VARIABLE MAP--(LOC)	--BLOCK--	--PROPERTIES--	--TYPE--	--SIZE--	--NAME--	--ADDRESS--	--BLOCK--	--PROPERTIES--	--TYPE--	--SIZE--
ATTRIB	18	/SSAUSE/ SAV	CHAR10		J	308	/SSAUSE/ SAV	INTEGER		60
CH	238	/SSAUSE/ SAV	CHAR10		K	318	/SSAUSE/ SAV	INTEGER		
ICONT	28	/NEX/	INTEGER		L	328	/LEVEL/	INTEGER		
IDATA	18	/NEX/	INTEGER		M	338	/LEVEL/	INTEGER		
IFADD	278	/LEVEL/	INTEGER		N	348	/LEVEL/	INTEGER		
IFIND	258	/LEVEL/	INTEGER		NDIFF	358	/LEVEL/	INTEGER		
II	278	/SSAUSE/ SAV	INTEGER		NLUIS	368	/SSAUSE/ SAV	CHAR10		16
INATT	328	/SSAUSE/ SAV	INTEGER		OBJCTV	378	/SSAUSE/ SAV	CHAR10		4
INARN	18	/LEVEL/	INTEGER		YATT	388	/P/	CHAR10		
ITOTL	28	/NEX/	INTEGER	20	USER	398				

--PROCEDURES--(LOC)	--TYPE--	--ARGS--	--CLASS--	--NAME--	--TYPE--	--ARGS--	--CLASS--
ATTSET	CHAR10	2	SUBROUTINE	PPI		0	SUBROUTINE
ATTI		1	FUNCTION	PRETO		0	SUBROUTINE
NEXT		0	SUBROUTINE	VALUE		0	SUBROUTINE
OBJECT		2	SUBROUTINE				

--STATEMENT LABELS--(LOC)	--DEF	--LABEL--ADDRESS--	--PROPERTIES--	--DEF	--LABEL--ADDRESS--	--PROPERTIES--	--DEF
1	138	5	3458	97	12	INACTIVE	87
2	238	10	INACTIVE	82	28	INACTIVE	43
3	1058	11	INACTIVE	81	99		56
4	1427						

--ENTRY POINTS--(LOC)	--NAME--	--ADDRESS--	--ARGS--
RDATT	58	0	

---STATISTICS---

PROGRAM-TEXT LENGTH 10238 - 571

OR LABELLED COMMON LENGTH 1658 - 117

OR SYNTAX USED 626608 - 25594

COMPILE TIME 8.249 SECONDS

SUBROUTINE RDSYSL

--VARIABLE MAP--(LO-A)
 --NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

CM 232B CHARZ1

--PROCEDURES--(LO-A)
 --NAME--TYPE--ARGS--CLASS--

WDSYS 0 SUBROUTINE
 DELSYS 0 SUBROUTINE
 NEU 0 SUBROUTINE
 PRETOT 0 SUBROUTINE

--STATEMENT LABELS--(LO-A)
 --LABEL-ADDRESS--PROPERTIES--DEF

1 7B 24

--ENTRY POINTS--(LO-A)
 --NAME--ADDRESS--ARGS--

RDSYSL 5B 2

--STATISTICS--

PROGRAM-UNIT LENGTH 241B 151
 CM STORAGE USED 60800 24000
 COMPILE TIME 0.002 SECONDS

```

1 SUBROUTINE RDITL
2
3 C
4 C
5 C
6 C
7 C
8 C
9 C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C

```

THIS ROUTINE ELICITS AN IDENTIFICATION FOR THE DATA
 STRUCTURE. IT MAY BE CALLED DIRECTLY VIA OPTION
 *** TIL ***. IT IS AUTOMATICALLY INVOKED WHEN USING
 OPTION *** NEW ***.

CALLED BY: DUMMY, INIT

VARIABLES
 USED: (NONE)
 MODIFIED: (NONE)

PRINT*,
 PRINT*, ENTER A TITLE FOR THIS DATA STRUCTURE...
 CALL CSET(1)
 RETURN
 END

```

--PROCEDURES--(LO-A)
--NAME--TYPE--ARGS--CLASS--
CSET 1 SUBROUTINE

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
RDITL 58 0

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH 438 35
CH STORAGE USED 86608 24068
COMPILE TIME 0.028 SECONDS

```


SUBROUTINE RCORS

```

58      IF (VAL(I)) .LT. 50.0) THEN
59      EXPX(I) = EXP(VAL(I))
60      ELSE
61      I = I + 1
62      IF (I) = 1
63      CONTINUE
64      VAL(I) = 0.0
65      VAL(I) = 0.0
66      VAL(I) = 0.0
67      VAL(I) = 0.0
68      VAL(I) = 0.0
69      VAL(I) = 0.0
70      VAL(I) = 0.0
71      VAL(I) = 0.0
72      VAL(I) = 0.0
73      VAL(I) = 0.0
74      VAL(I) = 0.0
75      VAL(I) = 0.0
76      VAL(I) = 0.0
77      VAL(I) = 0.0
78      VAL(I) = 0.0
79      VAL(I) = 0.0
80      VAL(I) = 0.0
81      VAL(I) = 0.0
82      VAL(I) = 0.0
83      VAL(I) = 0.0
84      VAL(I) = 0.0
85      VAL(I) = 0.0
86      VAL(I) = 0.0
87      VAL(I) = 0.0
88      VAL(I) = 0.0
89      VAL(I) = 0.0
90      VAL(I) = 0.0
91      VAL(I) = 0.0
92      VAL(I) = 0.0
93      VAL(I) = 0.0
94      VAL(I) = 0.0
95      VAL(I) = 0.0
96      VAL(I) = 0.0
97      VAL(I) = 0.0
98      VAL(I) = 0.0
99      VAL(I) = 0.0
100     VAL(I) = 0.0
101     VAL(I) = 0.0
102     VAL(I) = 0.0
103     VAL(I) = 0.0
104     VAL(I) = 0.0
105     VAL(I) = 0.0
106     VAL(I) = 0.0
107     VAL(I) = 0.0
108     VAL(I) = 0.0
109     VAL(I) = 0.0
110     VAL(I) = 0.0
111     VAL(I) = 0.0
112     VAL(I) = 0.0
113     VAL(I) = 0.0
114     VAL(I) = 0.0

```

SUBROUTINE REQNS

```

115 S5EXP=0.0
116 DO 30 I=1,5
117   S5=SSC*(ULU(I)-P0-B1XUL(I))**2
118   SSRT=SSC*(ULU(I)-B0ROOT-B1ROOT)*TX(I)**2
119   SSRT=SSC*(ULU(I)-B0SOR-B1SOR)*SORX(I)**2
120   SSILM=SSILM*(ULU(I)-B0LM-B1LM)*LX(I)**2
121   S5EXP=S5EXP+(ULU(I)-B0EXP-B1EXP)*EXP(I)**2
122   COT=1/ULU(I)
123   ULU(I)=COT*(SSC*SSRT+SSSOR,SSSELN,SSSEXP)
124   IF (ULU(I).EQ.0) THEN
125     CALL PSET(INDEX,1,0)
126     CALL PSET(INDEX,2,0)
127     CALL PSET(INDEX,3,0)
128     CALL PSET(INDEX,4,0)
129     ELSE IF (ULU(I).EQ.SSRT) THEN
130       CALL PSET(INDEX,1,B0ROOT)
131       CALL PSET(INDEX,2,B1ROOT)
132       CALL PSET(INDEX,3,SSRT)
133       CALL PSET(INDEX,4,1)
134       ELSE IF (ULU(I).EQ.SSOR) THEN
135         CALL PSET(INDEX,1,B0SOR)
136         CALL PSET(INDEX,2,B1SOR)
137         CALL PSET(INDEX,3,SSSOR)
138         CALL PSET(INDEX,4,1)
139       ELSE IF (ULU(I).EQ.SSELN) THEN
140         CALL PSET(INDEX,1,B0LN)
141         CALL PSET(INDEX,2,B1LN)
142         CALL PSET(INDEX,3,SSSELN)
143         CALL PSET(INDEX,4,2)
144       ELSE IF (ULU(I).EQ.SSEXP) THEN
145         CALL PSET(INDEX,1,B0EXP)
146         CALL PSET(INDEX,2,B1EXP)
147         CALL PSET(INDEX,3,SSSEXP)
148         CALL PSET(INDEX,4,2)
149       ENDIF
150       RETURN
151       END

```

--VARIABLE MAP--(I=1)	--NAME--	--ADDRESS--	--BLOCK--	--PROPERTIES--	--TYPE--	--SIZE--
DO	SSC	630	SSC	SAU	REAL	
EXP	SSC	730	SSC	SAU	REAL	
SELN	SSC	710	SSC	SAU	REAL	
ROOT	SSC	600	SSC	SAU	REAL	
EXP	SSC	670	SSC	SAU	REAL	
EXP	SSC	720	SSC	SAU	REAL	
EXP	SSC	700	SSC	SAU	REAL	
EXP	SSC	680	SSC	SAU	REAL	
EXP	SSC	660	SSC	SAU	REAL	
EXP	SSC	640	SSC	SAU	REAL	
EXP	SSC	620	SSC	SAU	REAL	
EXP	SSC	600	SSC	SAU	REAL	
EXP	SSC	580	SSC	SAU	REAL	
EXP	SSC	560	SSC	SAU	REAL	
EXP	SSC	540	SSC	SAU	REAL	
EXP	SSC	520	SSC	SAU	REAL	
EXP	SSC	500	SSC	SAU	REAL	
EXP	SSC	480	SSC	SAU	REAL	
EXP	SSC	460	SSC	SAU	REAL	
EXP	SSC	440	SSC	SAU	REAL	
EXP	SSC	420	SSC	SAU	REAL	
EXP	SSC	400	SSC	SAU	REAL	
EXP	SSC	380	SSC	SAU	REAL	
EXP	SSC	360	SSC	SAU	REAL	
EXP	SSC	340	SSC	SAU	REAL	
EXP	SSC	320	SSC	SAU	REAL	
EXP	SSC	300	SSC	SAU	REAL	
EXP	SSC	280	SSC	SAU	REAL	
EXP	SSC	260	SSC	SAU	REAL	
EXP	SSC	240	SSC	SAU	REAL	
EXP	SSC	220	SSC	SAU	REAL	
EXP	SSC	200	SSC	SAU	REAL	
EXP	SSC	180	SSC	SAU	REAL	
EXP	SSC	160	SSC	SAU	REAL	
EXP	SSC	140	SSC	SAU	REAL	
EXP	SSC	120	SSC	SAU	REAL	
EXP	SSC	100	SSC	SAU	REAL	
EXP	SSC	80	SSC	SAU	REAL	
EXP	SSC	60	SSC	SAU	REAL	
EXP	SSC	40	SSC	SAU	REAL	
EXP	SSC	20	SSC	SAU	REAL	
EXP	SSC	0	SSC	SAU	REAL	

SUBROUTINE REGRS		74/74 OPT-6		11/23/01 19.35.01		PAGE 4	
NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS
PSERT	768	/SSASUSE/	SAU	REAL		REAL	
SSGDR	788	/SSASUSE/	SAU	REAL		REAL	
SSGRN	448	/SSASUSE/	SAU	REAL		REAL	
SSWAXH	578	/SSASUSE/	SAU	REAL		REAL	
SULU	412	/SSASUSE/	SAU	REAL		REAL	
SULU2	548	/SSASUSE/	SAU	REAL		REAL	

PROCEDURES--(LO-A)		CLASS	
NAME	TYPE	ARGS	CLASS
ALOG	REAL	1	INTRINSIC
ANINI	REAL	1	INTRINSIC
EXP	GENERIC	1	INTRINSIC
PSET		3	SUBROUTINE

STATEMENT LABELS--(LO-A)		DEF	
NAME	ADDRESS	PROPERTIES	DEF
18	INACTIVE	DO-TERM	64
28	INACTIVE	DO-TERM	100
36	INACTIVE	DO-TERM	122

ENTRY POINTS--(LO-A)	
NAME	ADDRESS
REGRS	58

STATISTICS--	
PROGRAM-UNIT LENGTH	7238 * 467
OR LABELLED COMMON LENGTH	1078 * 71
OR STORAGE USED	686038 * 24980
COMPILE TIME	0.306 SECONDS

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE SPAN
C
C*****
C1 THIS ROUTINE CONDUCTS A MODE-BY-MODE CONSTRUCTION OF THE
C2 OBJECTIVE HIERARCHY. STARTING AT THE ROOT (OR DESIRED)
C3 MODE, ALL CHILDREN OF THE MODE ARE GENERATED. ONCE A
C4 SATISFACTORY SPAN OF CHILDREN IS ESTABLISHED, THE NEXT
C5 PARENT MODE IS CHOSEN IN A DEPTH-FIRST ORDER.
C6
C7 CALLED BY: DUMMY, INIT
C8
C9 VARIABLES
C10 USED: USER
C11
C12 MODIFIED: ANSWER, CH, CONTRL, ICONT, ICONT, IFADD,
C13 IFIND, IFINDT, IPRGN, LEVEL, LUL, NDEEP,
C14 NMODES, NMODN, TENSTR, U
C15
C16 *****
C17 COMMON/C/NODES, NDEEP
C18 COMMON/LEVEL/CHUL5, IPRGN(20), IFIND, NDIFF, IFADD, LUL, LEVEL(B0,3)
C19 COMMON/NEXT/ICONT - IDATA, ITOTL
C20 COMMON/CHILD/CONTRL
C21 COMMON/P/USER
C22 SAVE
C23 CHARACTER*16 USER, LABEL, ANSWER(16), TENSTR, OBJECTV(16), CH#1, UX1
C24 INTEGER CONTRL
C25 CALL PRETOT
C26 TENSTR = '
C27 WRITE(3, '(5(/), IX)')
C28 PRINT, 'SPANNING NODES: 'A'all '5'-SELECT'
C29 PRINT, '
C30 READ(3, '(A1)'ICH
C31 IF((CH.NE.'S').AND.(CH.NE.'A'))THEN
C32 PRINT, 'ONLY 'A' OR 'S' IS ALLOWED ', USER
C33 GO TO 1
C34 ENDF
C35 IF(ICH.EQ.'S')CALL PRENEX
C36 IFADD=8
C37 IF(ICONTR.NE.1)THEN
C38 WRITE(3, '(5(/), IX)')
C39 PRINT, 'ON YOU WISH TO BUILD A NEW TREE, 'USER,'? (Y/N)'
C40 PRINT, '
C41 READ(3, '(A1)'ICH
C42 IF((CH.NE.'N').AND.(CH.NE.'Y'))THEN
C43 PRINT, 'PLEASE INPUT 'Y' OR 'N', 'USER
C44 GO TO 8
C45 ENDF
C46 PRINT, '
C47
C48
C49
C50
C51
C52
C53
C54
C55
C56
C57

```

```

PRINTZ,'DO YOU WISH TO EV-PASS THE BETWEEN MODE CHECK?'
READ(X,(A)),J0
IF(CH.EQ.'N')THEN
ANSWER(1)='EXIT'
ELSE
LEVEL(1,1)=1
LEVEL(1,2)=1
LEVAL(1)=6
L=1
FIND=LEVEL(1,1)
LEAD=J
ICONT=1
NEALS=0
ENDIF
ENDIF
IF ICONT.NE.0)THEN
NOV=N+8
IF ICONT.EQ.FINE
CONTROL=0
PRINTZ,'ADDING DOWNLINKS TO NODE:'
WRITE(Z,(2013,X))IARMN(I),I-1,LUL
IF NNODES.GT.1)THEN
CALL OBJECT(LEVEL(LU),1,OBJECTU)
L=L+1
DO 20 I=1,4
WRITE(X,(1X,A10))OBJECTU(J),J=1,I,I+3
L=L+1+4
CONTINUE
ENDIF
PRINTZ,'
COUNT=ICOUNT+1
IF(ICOUNT.EQ.2)THEN
ANSWER(1)=TENSTR
ELSE
PRINTZ,'USER',WHAT IS THE NEXT SUBJECTIVE?'
PRINTZ,'USE NO MORE THAN TWO 20 CHARACTER LINES)'
PRINTZ,'
READ(Z,'(A10'))(ANSWER(1),I=1,2)
ENDIF
IF ((ANSWER(1).NE.TENSTR).AND.(ANSWER(1).NE.'DONE').AND.
((ANSWER(1).NE.'EXIT'))THEN
PRINTZ,'PLEASE CONTINUE'
PRINTZ,'
READ(Z,'(B10'))(ANSWER(1),I=1,10)
ENDIF
IF (ANSWER(1).EQ.'EXIT')RETURN
IF (CONTROL.LE.9).AND.(ANSWER(1).NE.TENSTR).AND.
((ANSWER(1).NE.'DONE'))THEN
CONTROL=CONTROL+1
MODEP=LUL+1
NODEN=NODEN+1
NODEN=NODEN+1
PRINTZ,'THE LAST SUBJECTIVE ENTERED IS:'
L=L+1
DO 10 I=1,4
WRITE(Z,(1X,A10'))(ANSWER(J),J=1,I,I+3)

```

```

10 10 11+4
11 CONTINUE
12 INNODES=OT.217*EN
13 CALL OBJECT(LEVEL(LVL,1),OBJECTU)
14 11+2
15 PRINT,'', 'WHICH IS SUBOBJECTIVE NUMBER ',MMX
16 PRINT,'FOR THE OBJECTIVE',
17 10 30 1-1,4
18 WRITE(2, '(1X,4A10)')(OBJECTU(J),J=1,11*3)
19 11+4
20 CONTINUE
21 30
22 ENDF
23 CALL ISET(IFINDT,LEADD,NNODES)
24 CALL ISET(NNODES,1,MMX)
25 CALL ISET(NNODES,2,-1)
26 CALL ISET(NNODES,3,-1)
27 CALL ISET(NNODES,4,IFINDT)
28 CALL ASSET(NNODES,1,0.0)
29 CALL ASSET(NNODES,2,0.0)
30 CALL OSET(NNODES,ANS,_R)
31 LEADD=3
32 IFINDT=NNODES
33 GO TO 4
34 ELSE
35 IF(LV.NE.'Y')THEN
36 CALL CHECK
37 ELSE
38 CALL NEXT
39 ENDF
40 IFADD=2
41 GO TO 5
42 ENDF
43 IF(LV.NE.'Y')THEN
44 WRITE(2, '(5(//),1X)')
45 CALL LEADD
46 RETURN
47 ENDF

```

---VARIABLE MAP---(LOW)		---PROPERTIES---		---TYPE---		---SIZE---	
NAME	ADDRESS	BLOCK	PROPERTIES	NAME	ADDRESS	BLOCK	PROPERTIES
ANSWER	2B		/S8A5USE/ SAV	J	52B		/S8A5USE/ SAV
CM	43B		/S8A5USE/ SAV	LEVEL	1B		/S8A5USE/ SAV
CONTROL	6B		/CHILD/	LEVEL	31B		/LEVEL/
I	59B		/S8A5USE/ SAV	LVL	30B		/LEVEL/
ICONT	6B		/MEX/	NDEEP	1B		/C/
ICOUNT	45B		/S8A5USE/ SAV	NDIFF	25B		/LEVEL/
IDATA	1B		/MEX/	NVLVS	6B		/LEVEL/
IPADD	27B		/LEVEL/	NMODS	3B		/C/
IPIND	25B		/LEVEL/	NMCM	45B		/S8A5USE/ SAV
IFINDY	47B		/S8A5USE/ SAV	OBJCTU	23B		/S8A5USE/ SAV
IF	51B		/S8A5USE/ SAV	PLNS:R	23B		/S8A5USE/ SAV
INON	1B		/LEVEL/	U	44B		/S8A5USE/ SAV
ITOTL	2B		/MEX/	USER	6B		/P/
CHAR210				CHAR210			
INTEGER				INTEGER			
CHAR211				CHAR211			
INTEGER				INTEGER			
CHAR212				CHAR212			
INTEGER				INTEGER			
CHAR213				CHAR213			
INTEGER				INTEGER			
CHAR214				CHAR214			
INTEGER				INTEGER			
CHAR215				CHAR215			
INTEGER				INTEGER			
CHAR216				CHAR216			
INTEGER				INTEGER			
CHAR217				CHAR217			
INTEGER				INTEGER			
CHAR218				CHAR218			
INTEGER				INTEGER			
CHAR219				CHAR219			
INTEGER				INTEGER			
CHAR220				CHAR220			
INTEGER				INTEGER			
CHAR221				CHAR221			
INTEGER				INTEGER			
CHAR222				CHAR222			
INTEGER				INTEGER			
CHAR223				CHAR223			
INTEGER				INTEGER			
CHAR224				CHAR224			
INTEGER				INTEGER			
CHAR225				CHAR225			
INTEGER				INTEGER			
CHAR226				CHAR226			
INTEGER				INTEGER			
CHAR227				CHAR227			
INTEGER				INTEGER			
CHAR228				CHAR228			
INTEGER				INTEGER			
CHAR229				CHAR229			
INTEGER				INTEGER			
CHAR230				CHAR230			
INTEGER				INTEGER			
CHAR231				CHAR231			
INTEGER				INTEGER			
CHAR232				CHAR232			
INTEGER				INTEGER			
CHAR233				CHAR233			
INTEGER				INTEGER			
CHAR234				CHAR234			
INTEGER				INTEGER			
CHAR235				CHAR235			
INTEGER				INTEGER			
CHAR236				CHAR236			
INTEGER				INTEGER			
CHAR237				CHAR237			
INTEGER				INTEGER			
CHAR238		</					

SUBROUTINE SPAN

--PROCEDURES--(LO-A)		--CLASS--		--NAME--TYPE--		--ARGS--		--CLASS--	
NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
ASET		SUBROUTINE		OCET		2		SUBROUTINE	
CHCK		SUBROUTINE		PREMEX		0		SUBROUTINE	
ISST		SUBROUTINE		PRETOT		0		SUBROUTINE	
NEXT		SUBROUTINE		RDATY		0		SUBROUTINE	
OBJECT		SUBROUTINE						SUBROUTINE	

--STATEMENT LABELS--(LO-A)		--DEF		--LABEL-ADDRESS--		--PROPERTIES--		--DEF	
NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS
1	13B	10	INACTIVE	DO-TERM	116				
2	50B	20	INACTIVE	DO-TERM	85				
3	130B	30	INACTIVE	DO-TERM	126				
4	224B								

--ENTRY POINTS--(LO-A)	
NAME	ADDRESS
SPAN	5B 0

--STATISTICS--	
PROGRAM-UNIT LENGTH	1105B - 591
CM LABELLED COMMON LENGTH	207B - 135
CM STORAGE USED	62600B - 25934
COMPILE TIME	0.286 SECONDS

```

1  SUBROUTINE STAT
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C

```

THIS ROUTINE DISPLAYS THE CHARACTERISTICS OF THE TREE
 FILE CURRENTLY IN USE. IT IS CALLED DIRECTLY VIA
 OPTION #88 STAT. IT IS INVOKED AUTOMATICALLY WHEN
 CONDUCTING THE BETWEEN-NODE CHECK.

CALLED BY: DUMV,CHECK

VARIABLES USED: NDEEP,NNODES,NSYS
 MODIFIED: (NONE)

COMMON/C,NNODES,NDEEP
 COMMON/NSYS1/NSYS

SAVE I2,(500,1X),
 PRINT*, 'CURRENT NUMBER OF NODES: ',NNODES, '(MAX 500)',
 PRINT*, 'CURRENT NUMBER OF LEVELS: ',NDEEP, '(MAX 20)',
 PRINT*, 'CURRENT NUMBER OF SYSTEMS: ',NSYS, '(MAX 50)',
 RETURN
 END

--VARIABLE MAP--(LO-A):

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

NDEEP 18 /C/ INTEGER
 NNODES 48 /C/ INTEGER
 NSYS 48 /C/ INTEGER

ENTRY POINTS--(LO-A):

--NAME--ADDRESS--APCS--

STAT SUBROUTINE STAT 74/74 OPT-0

--STATISTICS--

PROGRAM UNIT LENGTH 632 53
 CH LABELLED COMMON LENGTH 48 4
 CH STORAGE USED 50608 24960
 COMPILE TIME 9.032 SECONDS

```

1  SUBROUTINE TAPE
2
3  C*****
4  C
5  C THIS ROUTINE ALLOWS THE USER TO SELECT THE DESIRED
6  C DATA FILE. THE VARIABLE NTREE RESTRICTS THE CHOICE
7  C TO THREE POSSIBLE FILES.
8  C
9  C
10 C
11 C CALLED BY: DUMMY
12 C
13 C
14 C
15 C VARIABLES
16 C USED: USER
17 C
18 C MODIFIED: MNODES,NTAPE,NTREE
19 C
20 C*****
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C

```

COMMON / USER
COMMON / CTRL / ISAUD,NTAPE
SAVE
CHARACTER*10 USER
NTREE=3
MNODES=0
PRINT*, 'THE FOLLOWING INFORMATION WILL ALLOW YOU TO CHOOSE'
PRINT*, 'AN EXISTING (STORED) DATA FILE, OR TO CONSTRUCT A'
PRINT*, 'NEW ONE. ',USER,
PRINT*,
PRINT*, 'THE CURRENT TREE IS NUMBER ',NTAPE
CALL TSAVE
PRINT*,
PRINT*, 'WITH WHICH TREE WOULD YOU LIKE TO WORK. ',USER,
READ*, '(I1)',NTAPE
IF (NTAPE.EQ.1) OR (NTAPE.EQ.2) THEN
PRINT*, 'ONLY VALUES BETWEEN 1 AND ',NTREE, ' ARE ALLOWED.'
GO TO 1
ENDIF
CALL TLOAD
IF (MNODES.EQ.0) THEN
PRINT*, 'FILE ',NTAPE, ' HAS NO CURRENT TREE STRUCTURE. YOU ARE'
PRINT*, 'BEING TRANSFERRED TO OPTION 332 NEW 333.'
CALL INIT
ENDIF
RETURN
END

SUBROUTINE TAPER

```
--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
1SAUD      08 /CNTRL/
NNODES     28 /SABUAE/ SAV
NTAPE      18 /CNTRL/
NTRIEE     18 /SABUAE/ SAV
USER       02 /P/
          INTEGER
          INTEGER
          INTEGER
          INTEGER
          CHAR210
```

```
--PROCEDURES--(LO-A)
--NAME--TYPE--ARGS--CLASS--
INIT       0 SUBROUTINE
TL0AD      0 SUBROUTINE
TEAUC      2 SUBROUTINE
```

```
--STATEMENT LABELS--(LO-A)
--LABEL--ADDRESS--PROPERTIES--DEF
1 318 37
```

```
--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
TAPER      58 0
```

--STATISTICS--

```
PROGRAM-UNIT LENGTH      2128 - 138
CM LABELLED COMMON LENGTH  68 - 6
CM STORAGE USED          606008 - 24960
COMPILE TIME              0.066 SECONDS
```

```

1 SUBROUTINE VALLE
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

      THIS ROUTINE GENERATES THE DATA POINTS FROM WHICH THE
      INDIVIDUAL VALUE FUNCTIONS ARE ESTIMATED, AND CALLS
      ROUTINE REGRS TO DO THE ESTIMATION.

      CALLED BY: PFI,PDATT

      VARIABLES
      USED: NATT,XMID

      MODIFIED: CH,I,VAL,X

      COMMON/ATTR/MATT
      COMMON/VAL/VAL(5)
      SAVE
      CHARACTER CH,ATT*1810
      DO I=1,NATT
      CALL ASK(I,ATT(1,1),ATT(1,2),XMID)
      CALL(1):ATT(1,1)
      VAL(5)=ATT(1,2)
      VAL(3)=XMID
      X=XMID
      CALL ASK(I,ATT(1,1),X,XMID)
      VAL(2)=XMID
      CALL ASK(1,ATT(1,2),VAL(3),XMID)
      VAL(4)=XMID
      CALL REGRS(1)
      PRINT*,
      PRINT*, 'THE ABOVE YIELDS A VALUE FUNCTION FOR ',ATT(1,1)
      PRINT*, 'WITH PARAMETERS',
      PRINT*, '80',PARAM(1,1),
      PRINT*, 'SUP OF SQUARED ERROR-',PARAM(3,3)
      IF (PARAM(1,4).EQ.0.0) THEN
      PRINT*, '(LINEAR FORM)'
      PRINT*, 'VALUE=80+81*(ATTRIBUTE LEVEL)'
      ELSEIF (PARAM(1,4).EQ.1.0) THEN
      PRINT*, '(SQUARE-ROOT FORM)'
      PRINT*, 'VALUE=10+81*(ATTRIBUTE LEVEL)**.5'
      ELSEIF (PARAM(1,4).EQ.1.0) THEN
      PRINT*, '(SQUARED FORM)'
      PRINT*, 'VALUE=80+81*(ATTRIBUTE LEVEL)**2'
      ELSEIF (PARAM(1,4).EQ.-2.0) THEN
      PRINT*, '(LOGARITHMIC FORM)'
      PRINT*, 'VALUE=80+81*LN(ATTRIBUTE LEVEL)'
      ELSEIF (PARAM(1,4).EQ.2.0) THEN
      PRINT*, '(EXPONENTIAL FORM)'

```

SUBROUTINE VALUE

```

CB      PRINT, 'VALUE=90-BISEXP(A**ATTRIBUTE LEVEL)'
C9      CALL PLOT, R(1)
C1      PRINT, 'DOES THE ABOVE REPRESENTATION APPEAR REASONABLE? (Y/N)'
C2      PRINT, ' '
C3      READ, R(1), ICH
C4      IF (ICH.NE.'Y').AND.(ICH.NE.'N')) THEN
C5      PRINT, 'YOU MUST ENTER Y, N, OR *N*'
C6      GO TO 2
C7      ENDIF
C8      IF (CM.EQ. N) GO TO 1
C9      CONTINUE
C10     RETURN
C11     END

```

VARIABLE MAP--(LO-A)		--PROPERTIES--		--TYPE--		--SIZE--		--NAME--		--ADDRESS--		--BLOCK--		--PROPERTIES--		--TYPE--		--SIZE--	
CH	HT	IB	SSABUSE/ SAV	CHAR21	INTEGER	CHAR21	INTEGER	UCL	X	08	VAL/	48	SSABUSE/ SAV	08	VAL/	48	SSABUSE/ SAV	08	VAL/
1	10	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV
1	10	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV	08	SSABUSE/ SAV

--PROCEDURES--(LO-A)		--CLASS--		--NAME--		--TYPE--		--ARGS--		--CLASS--		--NAME--		--TYPE--		--ARGS--		--CLASS--	
NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE	NAME	TYPE
ASA	REAL	4	SUBROUTINE	PARAM	REAL	2	FUNCTION	2	FUNCTION	2	FUNCTION	2	FUNCTION	2	FUNCTION	2	FUNCTION	2	FUNCTION
ATT1	CHAR210	2	FUNCTION	PICTUR	REAL	1	FUNCTION	1	FUNCTION	1	FUNCTION	1	FUNCTION	1	FUNCTION	1	FUNCTION	1	FUNCTION

--STATEMENT LABELS--(LO-A)		--PROPERTIES--		--DEF	
1	148	29	61	70	70
2	1538	29	61	70	70
10	INACTIVE	DO-TERM			

--ENTRY POINTS--(LO-A)		--NAME--		--ADDRESS--		--ARGS--	
NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS	NAME	ADDRESS
VALUE	58	0					

--STATISTICS--	
PROGRAM-UNIT LENGTH	4078 - 211
CM LABELLED COMMON LENGTH	138 - 11
CM STORAGE USED	24000 - 24000
COMPILE TIME	0.121 SECONDS

```

1  OVERLAY=15.2,0)
2  PROGRAM _LOAD
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037

```

PROGRAM WULOAD

57

END

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
APRAY	0B	PRAY1		REAL	80
CH	0B	ONE		REAL	80
CM	0B	NEX		INTEGER	20
ICONT	0B	NEX		INTEGER	20
IFADD	0B	LEVEL		INTEGER	20
IFIND	0B	LEVEL		INTEGER	20
INPN	0B	LEVEL		INTEGER	20
ISAUD	0B	CTRL		INTEGER	20
ISIN	0B	PRAY2		INTEGER	20
ITOTL	0B	NEX		INTEGER	20

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CALL	0B	ONE		REAL	80
WULOD1	0B	NEX		INTEGER	20

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CALL	0B	ONE		REAL	80
WULOD1	0B	NEX		INTEGER	20

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CALL	0B	ONE		REAL	80
WULOD1	0B	NEX		INTEGER	20

STATISTICS

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CALL	0B	ONE		REAL	80
WULOD1	0B	NEX		INTEGER	20

```

1 SUBROUTINE CALC
2
3 *****
4
5 THIS ROUTINE CALCULATES THE CUMULATIVE HEIGHT OF EACH
6 MODE, AND THE ALTERNATIVE SYSTEM VALUES AT EACH
7 INTERIOR (NON-DATA) NODE.
8
9 *****
10
11 CALLED BY: WULOAD
12
13 *****
14
15 VARIABLES
16 USED: ICONT, IDATA, IFIND, LEVEL, LUL, NNODES, NSYS
17
18 MODIFIED: I, ISTOP, J, JJ, NDEEP, UNOLD
19
20 *****
21
22 COMMON/C/NNODES, NDEEP
23 COMMON/LEVEL/MLUS, INNRH(20), IFIND, NDIFF, IFADD, LUL, LEVEL(20,3)
24 COMMON/NEXT/ICONT, IDATA, I1OTL
25 COMMON/SYS1/NSYS
26
27 SAVE
28
29 IF(NNODES.LE.0)RETURN
30 PRINT*, 'INTERIOR TREE VALUES ARE BEING CALCULATED...'
31 NDEEP=0
32 CALL PRETOT
33 CALL ASET(1, 1-1, 0)
34 CALL ASET(1, 2, 1, 0)
35 CALL ASET(2, 1, 1, 0)
36 CALL ASET(2, 2, 1, 0)
37 IF(ICONT.EQ.0)THEN
38 CALL PRETOT
39 DO 5, I=1, NNODES
40 IF(ICONT.EQ.0)RETURN
41 IF(IDATA.NE.1)THEN
42 DO 4, JJ=1, NSYS
43 UNOLD=UNAV(IFIND, JJ)/GRAY(IFIND, 2)
44 CALL USET(IFIND, JJ, UNOLD)
45 CONTINUE
46 ENDIF
47 CALL NEXT
48 CONTINUE
49 ENDIF
50 IF(LUL.GT.NDEEP)NDEEP=LUL
51 UNOLD=UNAV(IFIND, 2)
52 IF(LUL-1).GE.1)
53 * UNOLD=UNAV(IFIND, 1)UNAV(LEVEL(LUL-1,1), 2)
54 CALL ASET(IFIND, 2, UNOLD)
55 IF(IDATA.NE.1)THEN
56 DO 10, I=1, NSYS
57 CALL USET(IFIND, J, 0.0)
58
59 *****
60
61
62 *****

```

```

88      ELSE
89      STOP,LVL-1
90      DO 20 I=1,IS-OP
91      DO 20 J=1,MSVS
92      UHOLD=UPRV(LEVEL(I,:),J)*ARRAY(IFIND,2)*URAY(IFIND,J)
93      CALL USET(LEVEL(I,:),UHOLD)
94      CONTINUE
95      ENDIF
96      CALL NEXT
97      GO TO 1
98      END

```

SUBROUTINE CALC

--STATISTICS--

	PROGRAM UNIT LENGTH
	3412
	CN LABELLED COMMON LENGTH
	1422
	CN STORAGE USED
	666923
	COMPILE TIME
	0.115 SECONDS

VARIABLE MAP--(LOC-A)			
--NAME--	--ADDRESS--	--BLOCK--	--SIZE--
I COUNT	0B	/SSASUSE/ SAV	
I DATA	0B	/NEX/	INTEGER
I FAD	1P	/NEX/	INTEGER
I FAND	27B	/LEVEL/	INTEGER
I FAND	28B	/LEVEL/	INTEGER
I I	1D	/SSASUSE/ SAV	INTEGER
I INRN	1B	/LEVEL/	INTEGER
I STOP	5B	/SSASUSE/ SAV	INTEGER
I TOTL	2B	/NEX/	INTEGER
I TOTL	49	/SSASUSE/ SAV	REAL
J LEVEL	2B	/SSASUSE/ SAV	INTEGER
K LEVEL	31B	/LEVEL/	INTEGER
L LEVEL	32B	/LEVEL/	INTEGER
MDEEP	1B	/LEVEL/	INTEGER
NDIFF	26B	/LEVEL/	INTEGER
NLOLS	0B	/LEVEL/	INTEGER
NMODS	0A	/C/	INTEGER
NSYS	0B	/SYS1/	INTEGER
NHOLD	2B	/SSASUSE/ SAV	REAL

--PROCEDURES--(LO-A)				--NAME--TYPE--ARGS--CLASS--				--NAME--TYPE--ARGS--CLASS--			
NAME	TYPE	ARGS	CLASS	NAME	TYPE	ARGS	CLASS	NAME	TYPE	ARGS	CLASS
ARRAY	REAL	2	FUNCTION	PRETOT		0	SUBROUTINE	ARRAY	REAL	2	FUNCTION
ACET		3	SUBROUTINE	URAY		3	SUBROUTINE	URAY		3	SUBROUTINE
NEXT		0	SUBROUTINE	USPT		0	SUBROUTINE	USPT		0	SUBROUTINE

```

-----STATEMENT LABELS---(LO-A)
-LABEL-ADDRESS-----PROPERTIES-----DEF
      1      20B
      1      INACTIVE      DO-TERM      37
      5      INACTIVE      DO-TERM      45
      10     INACTIVE      DO-TERM      57
      20     INACTIVE      DO-TERM      64

```

ENTRY POINTS--(L3-A)	NAME--ADDRESS--ARCS--	CALC	BR	Y


```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE SMOEC(NCROSS,IFIND,AIN)
C
C *****
C THIS ROUTINE ALLOWS THE USER TO ENTER THE RELATIVE WEIGHTS
C ACROSS THE SPAN IMPLICITLY BY USING A PAIRWISE-COMPARISON
C MATRIX. THE RELATIVE WEIGHTS ARE THE ELEMENTS OF THE
C GEOMETRIC PEAK VECTOR (ACROSS THE ROWS OF THE MATRIX)
C NORMALIZED TO SUM TO 1.
C *****
C CALLED BY: RDJF
C *****
C VARIABLES USED: IFIND,NCROSS,OBJCTU
C *****
C MODIFIED: AIN,ANSWER,I,I,IFINDT,J,K,KX,MATRIX,
C OBJCT,PROD,X1,X2
C *****
C
REAL AIN(20),MATRIX(20,20)
CHARACTER*16,OBJCT(15,16),OBJCTU(16)
INTEGER ANSWER
IFINDT=IRAY(IFIND,2)
CALL OBJECT(IFINDT,OBJCTU)
DO 20 I=1,NCROSS
DO 10 J=1,16
OBJCT(I,J)=OBJCTU(J)
IF(I.LT.NCROSS)THEN
IFINDT=IRAY(IFINDT,3)
CALL OBJECT(IFINDT,OBJCTU)
ENDIF
CONTINUE
DO 40 I=1,NCROSS
IF(I.LT.NCROSS)THEN
DO 30 J=I+1,NCROSS
PRINT*, 'USING THE SCALE:'
PRINT*, '1-9 AS DEFINED IN THE USER'S MANUAL.'
PRINT*, 'HOW IMPORTANT IS FACTOR ',I,'?'
I1=I
DO 50 K=1,4
WRITE(3, '(1X,4A10)',1)OBJCT(I,KK),KK=I1,I1+3)
I1=I1+4
CONTINUE
PRINT*, '
PRINT*, 'COMPARED TO FACTOR ',J,'?'
I1=I
DO 60 K=1,4
WRITE(3, '(1X,4A10)',1)OBJCT(J,KK),KK=I1,I1+3)
I1=I1+4
CONTINUE

```

SUBROUTINE GMVEC

```

58 PRINT*, 'ENTER THE NUMERATOR OF THE RATIO...'
59 READ*, (I1), ANSWER
60 X1=FLOAT(ANSWER)
61 PRINT*, 'ENTER THE DENOMINATOR OF THE RATIO...'
62 READ*, (I1), ANSWER
63 X2=FLOAT(ANSWER)
64 X2=X1/X2
65 MATRIX(I,J)=X1/X2
66 MATRIX(J,I)=X2/X1
67 CONTINUE
68 ENDIF
69 MATRIX(I,I)=1.0
70 PROD=1.0
71 DO 70 J=1, NCROSS
72 PROD=PROD*MATRIX(I,J)
73 AIN(I)=PROD/(1.0/PROD)
74 CONTINUE
75 RETURN
76 END

```

SUBROUTINE GMVEC

ENTRY POINTS--(LO-A)
 NAME--ADDRESS--ARGS--

GMVEC 58 3

--STATISTICS--

PROGRAM-UNIT LENGTH 16852 90
 CM STORAGE USED 60600 24960
 COMPILE TIME 0.161 SECONDS

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
VARIBLE MAP--(LO-A)					
AIN	1560B	3	DUMMY-ARG	INTEGER	400
ANSWER	1560B			REAL	
I	1562B			INTEGER	
IFIND	1561B	2	DUMMY-ARG	INTEGER	160
IFINDY	1570B			CHAR10	16
I1	1570B			CHAR10	
J	1570B			REAL	
K	1571B			REAL	
KK	1573B			INTEGER	
MATRIX	460B			REAL	
NCROSS	1			INTEGER	
OBJECT	1300B			CHAR10	
OBJECTU	1540B			CHAR10	
PROD	1641B			REAL	
X1	1577B			REAL	
X2	1600B			REAL	

--PROCEDURES--(LO-A)
 NAME--TYPE--ARGS--CLASS--

NAME	TYPE	ARGS	CLASS
FLAT	REAL	1	INTRINSIC
IPAY	INTEGER	2	FUNCTION
OBJECT			SUBROUTINE

--STATEMENT LABELS--(LO-A)
 LABEL-ADDRESS--PROPERTIES--DEF

LABEL	ADDRESS	PROPERTIES	DEF
10	INACTIVE	DO-TERM	34
20	INACTIVE	DO-TERM	39
30	INACTIVE	DO-TERM	52
40	INACTIVE	DO-TERM	76
50	INACTIVE	DO-TERM	50
60	INACTIVE	DO-TERM	57
70	INACTIVE	DO-TERM	73

```

SUBROUTINE PDV
C ***** THIS ROUTINE ELICITS THE ATTRIBUTE LEVELS OF THE ALTERNATIVE SYSTEMS ACROSS ALL THE DATA NODES. *****
C ***** CALLED BY: WU0D1 *****
C ***** VARIABLES USED: IFIND,NATT,NSYS,OBJCTU *****
C ***** MODIFIED: ANSWER,I,J,XMAX,KMIN *****
C ***** COMMON/ATTR/NATT *****
COMMON/LEVEL/MULDS,INRRY(28),IFIND,NDIFF,IFADD,LUT,LEVEL(29,3)
COMMON/NEV/CONT,IDATA,ITOTL
COMMON/SVS1/MSYS
SAVE
CHARACTER*16 LABEL-OBJECTU(16),SYSLSBL,ATT1
PRINT*,WE ARE AT THE DATA NODE:,
CALL OBJECT(IFIND,objctu)
I=1
DO 5 I=1,4
WRITE(*,'(IX,4A10)',O=objctu(K),K=1,I,1)+3)
II=II+4
CONTINUE
PRINT*,WHICH HAS THE ASSOCIATED ATTRIBUTE ,LABEL(IFIND)
DO 10 I=1,MSYS
PRINT*,THE CURRENT LEVEL OF THE ATTRIBUTE ,LABEL(IFIND)
PRINT*,IS ,XLEVEL(URAY(IFIND,I),LABEL(IFIND)), FOR SYSTEM'
PRINT,SYSLSBL(I)
DO 20 J=1,MATT
IF(LABEL(IFIND).EQ.ATT(J))THEN
PRINT,,THE RANGE OF THE ATTRIBUTE IS ,ATT(J,1),' TO ',ATT(J,2)
PRINT,*MINI(ATT(J,1),ATT(J,2))
XMIN=XMINI(ATT(J,1),ATT(J,2))
XMAX=XMAXI(ATT(J,1),ATT(J,2))
GO TO 1
ENDIF
ENDIF
PRINT,,
PRINT,"WHAT IS THE NEW LEVEL (REAL NUMBER)?",
READ*,(F10.0),ANSWER
IF((ANSWER.LT.XMIN).OR.(ANSWER.GT.XMAX))THEN
PRINT,,THE LEVEL OF ",ANSWER," IS OUT OF RANGE."
GO TO 1
ENDIF
ENDIF
```

SUBROUTINE PDV

```

10 ANSWER=VALU(ANSWER, LABEL(IFIND))
11 IF(ANSWER.LT.0.0) OR (ANSWER.GT.1.0) THEN
12 ANSWER=ANINI(ANSWER,1.0)
13 ANSWER=ANXI(ANSWER,1.0)
14 PRINT, 'THE INPUT ATTRIBUTE LEVEL CAUSES'
15 PRINT, 'THE VALUE GENERATED (BASED ON THE'
16 PRINT, 'ESTIMATED INDIVIDUAL VALUE FUNCTION)'
17 PRINT, 'TO BE OUTSIDE THE RANGE (0.0-1.0)'
18 PRINT, 'IN ORDER TO REMAIN IN THE PROPER RANGE.'
19 PRINT, 'YOUR INPUT VALUE IS BEING CHANGED TO'
20 PRINT, XLEVEL(ANSWER, LABEL(IFIND))
21 PRINT, 'IF THIS IS UNACCEPTABLE, USE **EXIT**'
22 PRINT, 'TO ADJUST THE VALUE FUNCTION, AFTER'
23 PRINT, 'EXITING THIS OPTION.'
24 ENDIF USE((IFIND,1,ANSWER)
25 CONTINUE
26 PRINT, '
27 PRINT, 'ENTER COMMENTS ON THESE ENTRIES.'
28 CALL CSET(IFIND)
29 RETURN
30 END

```

--STATEMENT LABELS--(LO-A)
 --LABEL-ADDRESS--PROPERTIES--DEF

1	1628	DO-TERM	51
5	INACTIVE	DO-TERM	35
10	INACTIVE	DO-TERM	74
20	INACTIVE	DO-TERM	50

--ENTRY POINTS--(LO-A)
 --NAME--ADDRESS--ARGS--

PDV	50	0
-----	----	---

--STATISTICS--

PROGRAM-UNIT LENGTH 5758 - 381
 CM LABELLED COMMON LENGTH 1628 - 114
 CM STORAGE USED 626008 - 25984
 COMPILE TIME 0.163 SECONDS

--VARIABLE MAP--(LO-A)
 --NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

ANSWER	278	/SSASUSE/ SAV	REAL	60
I	228	/LEVEL/	INTEGER	
LOMT	08	/NEV/	INTEGER	
LOATA	18	/NEV/	INTEGER	
LOADD	278	/LEVEL/	INTEGER	
LOIND	258	/LEVEL/	INTEGER	
LOINRM	18	/SSASUSE/ SAV	REAL	16
LOITOL	28	/NEV/	INTEGER	
J	248	/SSASUSE/ SAV	REAL	

--PROCEDURES--(LO-A)
 --NAME--TYPE--CLASS--ARGS--CLASS--

ANXI1	REAL	INTRINSIC	7	INTRINSIC
ANINI	REAL	FUNCTION	2	FUNCTION
ATT	REAL	FUNCTION	1	FUNCTION
CSET	CHAR12	FUNCTION	1	FUNCTION
LABEL	CHAR10	FUNCTION	1	FUNCTION

```
C  
C SUBROUTINE RDOT  
C *****  
C THIS ROUTINE ELICITS THE RELATIVE WEIGHTS OF ALL THE  
C MODES. THEY MAY BE INPUT DIRECTLY OR INDIRECTLY  
C (THROUGH THE USE OF A PAIRWISE-COMPARISON APPROACH).  
C INPUTS ARE NORMALIZED TO SUM TO UNITY ACROSS EACH SPAN.  
C *****  
C CALLED BY: WUOLDI  
  
C VARIABLES USED: LEVEL,OBJECTV  
  
C MODIFIED: AIMCM,I,IFIND,IFINDT,I,I,ITOTL,J,K,  
C NCROSS,ZNCRH  
C *****  
C COMMON/CNTL/ISAVD,NTAPE  
C COMMON/LEVEL/NULUS,INTRM(28),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)  
C COMMON/MEX/ICONT,IDATA,ITOTL  
C SAVE  
C CHARACTERZ@ OBJECTV(16),CH#1  
C REAL AIM(28)  
C I=J-LEVEL(LUL,1)  
C IFINDT-IWAY(IFIND,2)  
C NCROSS=-1  
C PRINT*,WE ARE WEIGHING THE NODE SET:  
C CALL OBJECT(IFINDY,OBJCTV)  
C I,I+1  
C DO 5 J=1,4  
C WRITE(2,'(IX,4A10)',IOBJCTV(K),K=II,II+3)  
C II=II+4  
C CONTINUE  
C PRINT*. THE ABOVE OBJECTIVE IS FACTOR 1'  
C IF IND-T IRAY(IFINDT,3)  
C IF(IFINDT.GT.@)THEN  
C CALL OBJECT(IFINDT,OBJCTV)  
C II=1  
C DO 10 J=1,4  
C WRITE(2,'(IX,4A10)',IOBJCTV(K),K=II,II+3)  
C II=II+4  
C CONTINUE  
C NCROSS=NCROSS+1  
C PRINT*,THE ABOVE OBJECTIVE IS FACTOR ',NCROSS  
C GO TO 1  
C ENDIF  
C PRINT*,  
C PRINT*,DO YOU WISH TO ENTER THE RELATIVE WEIGHTS DIRECTLY.'  
C PRINT*,'(V,M)' ?
```

SUBROUTINE RDUT

```

58 READL, (A1), ICH
59 IF (CH.NE.'V').AND.(CH.NE.'N'))THEN
60 PRINT, 'ENTER 'Y' OR 'N'.'
61 GO TO 2
62 ENDIF
63 IF (CH.EQ.'N')THEN
64 CALL GRUEC(MCROSS, IFIND, AIN)
65 ELSE
66 PRINT, 'ENTER THE (UNNORMALIZED) WEIGHTS.'
67 DO 20 I=1, MCROSS
68 PRINT, 'WHAT IS THE WEIGHT FOR FACTOR ', I
69 PRINT, ' '
70 READ, AIN, I
71 IF (AIN(1).EQ.0.0)GO TO 4
72 CONTINUE
73 IF (AIN(1).EQ.0.0)THEN
74 :TOTL=2
75 RETURN
76 ENDIF
77 ENDIF
78 IF (MCROSS.EQ.1)CALL ASET(IRAY(IFIND, 2), 1.0, 0)
79 ZNORM=SUM(MCROSS, AIN)
80 IF (ZNORM.LT.0.0)RETURN
81 IF (ZNORM.GT.0.0)CALL DUIDE(MCROSS, AIN, ZNORM)
82 PRINT, 'NORMALIZED: ', ((FIX(100.0*AIN(I)+.5)), 1=1, MCROSS)
83 PRINT, 'ARE YOU HAPPY WITH THESE RELATIVE WEIGHTS? (Y/N)'
84 PRINT, ' '
85 READ, (A1), ICH
86 IF (CH.NE.'Y').AND.(CH.NE.'N'))THEN
87 PRINT, 'PLEASE ENTER 'Y' OR 'N'.'
88 GO TO 3
89 ENDIF
90 IF (CH.EQ.'N')GO TO 2
91 IFIND=IRAY(IFIND, 2)
92 DO 11 I=1, MCROSS
93 CALL ASET(IFINDT, 1, AIN(I))
94 IFINDT=IRAY(IFINDT, 3)
95 CONTINUE
96 PRINT, 'ENTER COMMENTS ON THESE WEIGHTS'
97 CALL CSET(IFIND)
98 RETURN
99 END
100

```

---VARIABLE MAP---(LO-a)			---PROPERTY---BLOCK			---PROPERTY---BLOCK			---TYPE---BLOCK			---SIZE		
NAME	ADDRESS	BLOCK	NAME	ADDRESS	BLOCK	NAME	ADDRESS	BLOCK	NAME	ADDRESS	BLOCK	NAME	ADDRESS	BLOCK
AIN	228	/SSABUSE/ SAU	IMMM	18	/LEVEL/	ICH	48	/CTRL/	INTEGER			20		
CH	218	/SSABUSE/ SAU	ISAND	23	/LEN/	ICH	48	/CTRL/	INTEGER					
I	518	/SSABUSE/ SAU	ITOTL	518	/SSABUSE/ SAU	ICH	48	/CTRL/	INTEGER					
ICONT	48	/NEX/	J	528	/SSABUSE/ SAU	ICH	48	/CTRL/	INTEGER					
IPDATA	278	/LEVEL/	K	528	/SSABUSE/ SAU	ICH	48	/CTRL/	INTEGER					
IPADD	258	/LEVEL/	LEVEL	528	/SSABUSE/ SAU	ICH	48	/CTRL/	INTEGER					
IFIND	468	/SSABUSE/ SAU	LUL	308	/LEVEL/	ICH	48	/CTRL/	INTEGER					
IFINDT	508	/SSABUSE/ SAU	MCROSS	478	/SSABUSE/ SAU	ICH	48	/CTRL/	INTEGER					
II			NDIFF	268	/LEVEL/	ICH	48	/CTRL/	INTEGER					

SUBROUTINE RDWT 7474 OPT=0 11/23/81 19.35.01 PAGE 3
 -NAME-----BLOCK-----PROPERTIES-----TYPE-----SIZE
 NCULS 00 /LEVEL/ INTEGER 18 /SSABUSE/ SAV CHARS10 16
 HTAPE 10 /CTRL/ INTEGER 548 /SSABUSE/ SAV REAL

-PROCEDURES--(LO-A)
 -NAME-----TYPE-----ARGS-----CLASS-----
 ASET SUBROUTINE 3
 CSET SUBROUTINE 1
 DUIDE SUBROUTINE 3
 LRVEC SUBROUTINE 3

-STATEMENT LABELS--(LO-A)
 -LABEL ADDRESS-----PROPERTIES-----DEF
 1 66B 43
 2 144B 55
 3 308B 83
 4 227B 73

-ENTRY POINTS--(LO-A)
 -NAME-----ADDRESS-----ARGS-----
 RDWT 5B 0

-STATISTICS--
 PROGRAM-UNIT LENGTH 6318 : 402
 CM LABELLED COMMON LENGTH 2878 : 135
 CM STORAGE USED 586003 : 24960
 COMPILE TIME 9.153 SECONDS


```

1  SUBROUTINE LWLOAD:(OPT)
2
3  C*****
4  C THIS ROUTINE DIRECTS THE INPUT OF RELATIVE
5  C WEIGHTS OR ATTRIBUTE LEVELS.
6  C*****
7  C
8  C CALLED BY: LWLOAD
9  C
10 C
11 C VARIABLES
12 C   USED: ICONT,OPT
13 C   MODIFIED: CH,CH1,LABEL
14 C*****
15 C
16 C COMMON/NEX/ICONT, IDATA,ITOTL
17 C
18 C SAVE
19 C CHARACTER CH,CH1,OPT,LABEL$10
20 C IF(OPT.EQ.'U') THEN
21 C   LABEL='WEIGHTS'
22 C ELSE
23 C   LABEL='VALUES'
24 C ENDIF
25 C PRINT*,LABEL,' I WILL SELECT'
26 C PRINT*,(A1)' ICH
27 C READ*,(A1)' ICH
28 C IF(CH.NE.'A') AND (CH.NE.'S') THEN
29 C   PRINT*, 'I AM SORRY, BUT I CAN ONLY ACCEPT 'A' OR 'S'.'
30 C   GO TO 1
31 C ENDIF
32 C PRINT*, ' '
33 C CALL PRETOT
34 C IF(CH.EQ.'S') CALL PRENEX
35 C IF(ICONT.EQ.0) THEN
36 C   PRINT*, 'NEW ENTERED IS INVALID.'
37 C   PRINT*, '(PRESS ANY LETTER TO CONTINUE)'
38 C   PRINT*, ' '
39 C   READ*, (A1)' ICH1
40 C ENDIF
41 C IF((ICONT.EQ.0).AND.(CH.EQ.'S')) GO TO 2
42 C IF((IDATA.EQ.0).AND.(OPT.EQ.'U')) CALL RWUT
43 C IF((IDATA.EQ.1).AND.(OPT.EQ.'U')) CALL RWU
44 C IF((ITOTL.EQ.1)) CALL NEXT
45 C IF((ICONT.NE.0).AND.(ITOTL.EQ.1)) GO TO 3
46 C IF((ITOTL.EQ.0) GO TO 2
47 C RETURN
48 C END
49
50
51
52
53
54
55

```

SUBROUTINE WULOD:

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ITOTL	28	/NEX/		INTEGER	
LABEL	33	/SEASUSE/	SAU	CHAR*10	
OPT	1	DUMMY-ARG		CHAR*1	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

STATISTICS--

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
CHAR*1				CHAR*1	
CHAR*1				CHAR*1	
INTEGER				INTEGER	
INTEGER				INTEGER	

```

1  OVERLAY('XFILE,3,6)
2  PROGRAM RUN
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

```

      THIS ROUTINE IS THE ENTRY POINT FOR OVERLAY(3,6).  IT
      ACTS AS A DRIVER FOR THE DISPLAY/REVIEW OPTIONS.

      CALLED BY:  (NONE)

      VARIABLES
      USED:  CND, IDATA
             MODIFIED:  CH

      COMMON/ATTR/MATT
      COMMON/C/NODES,NDEEP
      COMMON/CTRL/ISAUD,NTAPE
      COMMON/LEVEL/NCULS,INRM(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
      COMMON/TEXT/CONFI, IDATA,ITOTL
      COMMON/ONE/CRD
      COMMON/P/USER
      COMMON/ARRAY1/ARRAY(80)
      COMMON/ARRAY2/ISIM,NLOUD
      COMMON/SYS1/NSYS
      SAVE
      CHARACTER*3 CND,CHS1
      CALL PRETOT
      IF(CND.EQ.'DIS')THEN
        WRITE(S,'(5(/),1X)')
        PRINT*, 'ENTER NODE TO BE DISPLAYED...'
        CALL MODIN
        IF(IDATA.EQ.1)THEN
          PRINT*, 'NODE IS A DATA NODE, AND CANNOT BE DISPLAYED'
          PRINT*, 'PRESS ANY LETTER TO CONTINUE',
            PRINT*, '?'
          READ(S,'(A1)')CH
          GO TO 1
        ELSE
          CALL PRETOT
          WRITE(S,'(5(/),1X)')
          PRINT*, 'HOW MUCH DO YOU WANT TO REVIEW...'
          PRINT*, 'ALL  SELECT'
          PRINT*, '?'
          READ(S,'(A1)')JCH
          IF((CH.NE.'A') .AND. (CH.NE.'S'))THEN
            PRINT*, 'ENTER "A" OR "S"'
            GO TO 3
          ENDIF
        ENDIF
      ENDIF

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57

PROGRAM NUM

```
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
ANSI  
X  
Z  
ENDIF  
IF(CMD.EQ.'S')CALL PRENEX  
ENDIF  
PRINT*, 'IF ANY MODIFICATIONS HAVE BEEN MADE TO THE TREE'  
PRINT*, 'SINCE IT HAS BEEN CALCULATED, NUMERICAL VALUES'  
PRINT*, 'WILL BE INCORRECT.'  
PRINT*, '(PRESS ANY LETTER TO CONTINUE)'  
PRINT*, ' '  
READ(X, '(R1)')  
IF(CMD.EQ.'DIS')THEN  
CALL DISPLA  
ELSEIF(CMD.EQ.'NUM')OR(CMD.EQ.'REV')THEN  
CALL NUMREV  
ENDIF  
RETURN IN MAIN PROGRAM -- ACTS AS END  
END
```

PROGRAM NUM

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--

NUM 08 0

--STATISTICS--

PROGRAM-UNIT LENGTH 09 0
CM LABELLED COMMON LENGTH 2648 180
CM STORAGE USED 606009 24060
COMPILE TIME 0.005 SECONDS

1 ANSI ERROR IN NUM

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ARRAY	08	/RRAY1/		REAL	80
CM	08	/SASUSE/	SAS	CHARX1	
CM	08	/ONE/		CHARX3	
ICONT	08	/HEX/		INTEGER	
IDATA	08	/HEX/		INTEGER	
IFADD	08	/LEVEL/		INTEGER	
IFIND	08	/LEVEL/		INTEGER	
INRW	08	/LEVEL/		INTEGER	
ISAD	08	/CTRL/		INTEGER	
ISIN	08	/RRAY2/		INTEGER	
ITOTL	08	/HEX/		INTEGER	
LEVEL	08	/LEVEL/		INTEGER	60
LVL	08	/LEVEL/		INTEGER	
MATT	08	/ATTR/		INTEGER	
HDEEP	08	/C/		INTEGER	
NDIFF	08	/LEVEL/		INTEGER	
NLOUD	08	/RRAY2/		INTEGER	
NLUIS	08	/LEVEL/		INTEGER	
NRDES	08	/C/		INTEGER	
NSYS	08	/S/S1/		INTEGER	
NRAPE	08	/CTRL/		INTEGER	
USER	08	/P/		REAL	

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56
SUBROUTINE DISPLA
C*****
C THIS ROUTINE GENERATES A TABULAR DISPLAY OF THE
C INFORMATION ASSOCIATED WITH A SELECTED NODE.
C*****
C CALLED BY: NMF
C*****
C VARIABLES USED: IDATA, IFIND, NSYS, OBJCTU
C MODIFIED: DUM, I, IFINDT, II, IJ, INDEX, K, NCROSS,
C TSYS, VAL, UT
C*****
C*****
COMMON/CTRL/ISAUD, NTAPE
COMMON/LEVEL/HLULS, IMMPH(20), IFIND, NDIFF, IFADD, LUL, LEVEL(20,3)
COMMON/TEXT/CONT, IDATA, ITOTL
COMMON/SYS1/RSYS
CHARACTER*18 OBJCTU(16), DUR, SYSLEL, TSYS(4)
REAL VAL(4)
IF(IDATA.EQ.1) THEN
  PRINT*, 'A DATA NODE MAY NOT BE DISPLAYED.'
  PRINT*, ''
  RETURN
ENDIF
PRINT*, 'THE PARENT OBJECTIVE IS:'
CALL OBJECT(IFIND, OBJCTU)
II=1
DO 10 I=1,4
  WRITE(3, '(IX,4I9)')(OBJCTU(K), K=II, II+3)
  II=II+4
CONTINUE
NCROSS=1
IFINDT=IFAV(1:IND,2)
DO 20 I=1,10
  IF(I*AV(IFINDT,3).GT.0) THEN
    NCROSS=NCROSS+1
    IFINDT=IFAV(IFINDT,3)
  ENDIF
CONTINUE
PRINT*, 'IT HAS', NCROSS, ' SUBOBJECTIVES (FACTORS):'
IFINDT=IFAV(IFIND,2)
DO 40 I=1, NCROSS
  CALL OBJECT(IFINDT, OBJCTU)
  PRINT*, 'FACTOR', I, ':'
  II=1
  DO 30 J=1,4
    10
    20
  END

```

[illegible][illegible]

```

--ENTRY POINTS--(LOAR)
--NAME--ADDRESS--ARGJ---
DISPLA 58 0

```


SUBROUTINE DSPLT

```

58 CONTINUE
59 PRINT, 'FACTOR ', MOUNT, DUM, DSPL
60 GO TO 1
61 ENDIF
62 PRINT, 'BRANCH ',
63 PRINT, ' ',
64 PRINT, 'LEGEND',
65 PRINT, ' ',
66 INDEX=INDEX+2
67 DO 40 J=1, INDEX
68 PRINT, SYMBOL, ALFA(J), ' IS SYSTEM', SYSLBL(J),
69 PRINT, SYMBOL, ALFA(J+1), ' SYSTEM', SYSLBL(J+1)
70 IF (J+1) 2
71 CONTINUE
72 IF (2) INDEX=1, INDEX=THEN
73 PRINT, SYMBOL, ALFA(INDEX), ' SYSTEM', SYSLBL(INDEX)
74 PRINT, ' ',
75 PRINT, ' ',
76 RETURN
77 END
78
--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
ALFA 18 /SSAUSE/ SAV 50
DSPL 68 /SSAUSE/ SAV 51
DUM 148 /SSAUSE/ SAV
I 208 /SSAUSE/ SAV
ICON 88 /NEX/
IDATA 18 /NEX/
IF20 278 /LEVEL/
IFIND 258 /LEVEL/
IFINDT 168 /SSAUSE/ SAV
IFROM 158 /SSAUSE/ SAV
II 258 /SSAUSE/ SAV
IINDEX 268 /SSAUSE/ SAV
INDEX 228 /SSAUSE/ SAV

--PROCEDURES--(LO-A)
--NAME--TYPE--ARGS--CLAS--
IRAY INTEGER 2 FUNCTION
NING INTEGER 7 INTRINSIC
SYSLBL CHAR*16 1 FUNCTION
URAY REAL 2 FUNCTION

```

SUBROUTINE DSPLT

```

--STATEMENT LABELS--(LO-A)
--LABEL--ADDRESS--PROPERTIES--DEF
1 228 DO-TERM 41
10 INACTIVE DO-TERM 46
20 INACTIVE DO-TERM 48
30 INACTIVE DO-TERM 58
40 INACTIVE DO-TERM 72

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
DSPLT 58 0

--STATISTICS--
PROGRAM-UNIT LENGTH 4038 - 259
CM LABELLED COMMON LENGTH 1628 - 114
CM STORAGE USED 68608 - 24960
CM COMPILATION TIME 0.168 SECONDS

--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE
18 /LEVEL/ INTEGER 20
248 /SSAUSE/ SAV INTEGER
68 /CTRL/ INTEGER
28 /NEX/ INTEGER
228 /SSAUSE/ SAV INTEGER
318 /LEVEL/ INTEGER
308 /LEVEL/ INTEGER
378 /SSAUSE/ SAV INTEGER
268 /LEVEL/ INTEGER
68 /LEVEL/ INTEGER
68 /SYSI/ INTEGER
118 /CTRL/ INTEGER
218 /SSAUSE/ SAV REAL

```

```

1  SUBROUTINE NCUPG
2  C
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C

```

THIS ROUTINE PROVIDES A HEADER FOR EACH NODE IN THE
 *** NUK *** OPTION.

CALLED BY: NUKREV

VARIABLES
 USED: (NONE)
 MODIFIED: (NONE)

WRITE(2, '(S(/),1X)')
 PRINT*,
 CALL CRAY(1)
 PRINT*,
 PRINT*, 'NODE REFERENCE NUMBER(AND OBJECTIVE):'
 RETURN
 END

```

--PROCEDURES--(LO-A)
--NAME-----TYPE-----ARGS-----CLASS-----
CRAY          1      SUBROUTINE

```

```

--ENTRY POINTS--(LO-A)
--NAME-----ADDRESS-----ARGS-----

```

```

NCUPG          59      0

```

```

--STATISTICS--

```

```

PROGRAM-UNIT LENGTH      692 - 48
CM STORAGE USED          68608 - 24063
COMPILE TIME              0.024 SECONDS

```

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57
SUBROUTINE NUMREV
C *****
C THIS ROUTINE CONDUCTS A NODE-BY-NODE REVIEW OF THE
C TREE. THE $$$ NUM $$$ OPTION GIVES NUMERICAL
C INFORMATION IN ADDITION TO THE INFORMATION PROVIDED
C BY THE $$$ REV $$$ OPTION.
C *****
C CALLED BY: NUM
C *****
C VARIABLES
C   USED: CMD,ICONT,IFIND,INNER
C   MODIFIED: CH,II,IJ,INDEX,J,K,TSYS,VAL
C *****
COMMON/LEVEL/MLULS,INNER(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
COMMON/ONE/CMD
COMMON/SYS/NSYS
SAVE
CHARACTER*40 CH$1,CMD$3,OBJECTU(16),SYSLBL,TSYS(4)
REAL VAL(4)
IF(CMD.EQ.'REV')THEN
  WRITE(2, '(5I/,1X)')
  CALL CRAY(1)
ELSE
  CALL NEUPG
ENDIF
WRITE(1, '(20I3)',(INNER(I),I=1,LUL)
CALL OBJECT(IFIND,OBJECTU)
II=1
DO 10 J=1,4
  WRITE(2, '(1X,4A10)',(OBJECTU(K),K=II,II+3)
  II=II+4
CONTINUE
IF(CMD.EQ.'NUM')THEN
  PRINT2, 'RELATIVE WEIGHT: ',ARAY(IFIND,1)
  PRINT2, 'CUMULATIVE WEIGHT: ',ARAY(IFIND,2)
  PRINT2, 'SYSTEM VALUES:'
  II=1
  INDEX=NSYS/4
  DO 20 J=1,INDEX
    IJ=1
    DO 10 K=1,II+3
      TSYS(IJ)=SYSLBL(K)
      VAL(IJ)=100.0*UMAY(IFIND,K)
      IJ=IJ+1
    CONTINUE
  END

```

```

WRITE('',(X,X,4(A10,EX)),'(TSYS(K),K=1,4)
WRITE('',(X,X,4(F10,EX)),'(VAL(K),K=1,4)
CONTINUE
IF(4*INDEX(LT,NSYS)THEN
  IJ=1
  DO 21 K=INDEX+1,NSYS
    TSYS(IJ)=SYSLBL(K)
    VAL(IJ)=100.0*UEXY(IFIND,K)
    IJ=IJ+1
  CONTINUE
  WRITE('',(X,X,4(A10,EX)),'(TSYS(K),K=1,IJ-1)
  WRITE('',(X,X,4(F10,EX)),'(VAL(K),K=1,IJ-1)
  ENDIF
  PRINT*,' '
  CALL CRAY(:IFIND)
  ELSE
    PRINT*,' '
  ENDF
  CALL NEXT
  PRINT*,'(PRESS ANY LETTER TO CONTINUE (EXCEPT
  PRINT*,'(PRESS 'E' TO EXIT)'
  PRINT*,' '
  READ('C',(A1))'C'
  IF(LOC(NE,'E').AND.(ICONT.NE.0))GO TO 1
  RETURN
END

```

STATEMENT LABELS--(LO=0)	PROPERTIES----	REF
1	DO-TERM	31
10	INACTIVE	37
19	INACTIVE	25
20	INACTIVE	61
21	INACTIVE	63

```
--ENTRY POINTS--(LO=A)
--NAME--ADDRESS--ARGS---
```

--501511815--

PROGRAM-UNIT LENGTH	5553	•	385
CM LABELLED COMMON LENGTH	1708	•	120
CM STORAGE USED	60600	•	24960
COMPILE TIME	6.178		SECONDS

VARIABLE MAP--(10*A)		NAME--ADDRESS--BLOCK--		PROPERTIES--TYPE--		SIZE	
CH	12	/SSASUSE/	SAU	CHAR11			
CMD	03	/ONE/		CHAR13			
1	328	/SSASUSE/	SAU	INTEGER			
ICONT	408	/SSASUSE/	UND/SAU	INTEGER			
IFADD	278	/LEVEL/		INTEGER			
IFIND	258	/LEVEL/		INTEGER			
11	332	/SSASUSE/	SAU	INTEGER			
11	378	/SSASUSE/	SAU	INTEGER			

PROCEDURES--(LOA)			SUBROUTINE		
NAME	TYPE	ARGS	NAME	TYPE	ARGS
ARRAY	REAL	2	OBJECT	CHAR10	2
CRAY		1	SYSTEM		1
NEWPG		6	WRAY	REAL	2
NEXT		0			

```

1  OVERLAY(XFILE,4,0)
2  PROGRAM SENSU
3
4
5
6
7
8  THIS PROGRAM IS THE ENTRY POINT FOR OVERLAY(4,0).
9  IT DIRECTS THE VARIOUS SENSITIVITY ANALYSIS OPTIONS.
10
11  CALLED BY: (NONE)
12
13  VARIABLES
14      USED: (NONE)
15
16      MODIFIED: CH,CHI,SENS
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31  COMMON/ATTR/MATT
32  COMMON/CANODES/NDEEP
33  COMMON/CNTRL/ISAD,NBASE
34  COMMON/LEVEL/NLULS,INNRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
35  COMMON/NEXT/ICONT,IDATA,ITOTL
36  COMMON/CNE/CND
37  COMMON/P/USER
38  COMMON/ARRAY1/ARRAY(80)
39  COMMON/ARRAY2/ISIN,NLQUD
40  COMMON/SEN1/ATTNAM,SENS,SYSNAM
41  COMMON/SYS1/NSYS
42
43  SAVE
44  CHARACTER*20 SENSIS,SYSNAM,CHI1,CHI1,ATTNAM
45  WRITE(*, '(51,/,X)')
46  PRINT*
47  PRINT*,'CHOOSE YOUR OPTION:'
48  PRINT*,' C/JUNT R/ELUT'
49  PRINT*,' ?'
50  READ(*, '(A1)')CH
51  IF((CH.NE.'C').AND.(CH.NE.'R').AND.(CH.NE.'L').AND.
52  * (CH.NE.'E').AND.(CH.NE.'S'))THEN
53  PRINT*,'PLEASE ENTER 'C','R','L','S', OR 'E'
54  GO TO 1
55  ENDIF
56
57  CALL PRETOT
58  IF(CH.NE.'E')THEN
59  IF(CH.EQ.'C')THEN
60  SENS='CJUNT'
61  ELSEIF(CH.EQ.'R')THEN
62  SENS='RELUT'
63  ELSEIF(CH.EQ.'S')THEN
64  SENS='SYSNAM'
65  ELSE
66  SENS='SYSNAM'
67  ENDIF
68
69  LLEVEL = SYSTEM
70  EXIT

```

PROGRAM SENSU

```

50 SENS='LEVEL'
51 ENDIF
52 IF(SENS.NE.'SYGTH')THEN
53 CALL DETNOD
54 ELSE
55 CALL SYSEN
56 ENDIF
57 CALL COMPUT
58 WRITE(2,(5(1),1X))
59 PRINT(2,SENS,' ANALYSIS: TABULAR GRAPHICAL EXIT'
60 PRINT(2,'
61 READ(2,(A1))CH1
62 IF((CH1.NE.'T').AND.(CH1.NE.'S').AND.(CH1.NE.'E'))THEN
63 PRINT(2,'ENTER 'T','S', OR 'E'
64 GO TO 2
65 ENDIF
66 IF(CH1.EQ.'T')THEN
67 CALL TABDIS
68 ELSEIF(CH1.EQ.'S')THEN
69 CALL GRAPH
70 ENDIF
71 IF(CH1.NE.'E')GO TO 2
72 RETURN
73 END
74
75
76
77
78
79
80
81
82

```

PROGRAM SENSU

```

--STATEMENT LABELS--(LO-A)
--LABEL-ADDRESS--PROPERTIES--DEF
1 08 37
2 08 66

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARGS--
SENSU 08 0

```

--STATISTICS--

```

PROGRAM-UNIT LENGTH 08 0
CM LABELLED COMMON LENGTH 2708 184
CM STORAGE USED 606008 24960
CMPILE TIME 0.114 SECONDS

```

```

--VARIABLE MAP--(LO-A)
--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE

```

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ARRAY	08	ARRV1	REAL	REAL	80
ATTNAR	08	SEMI	CHAR10	CHAR10	80
CH	08	SEABUSE	CHAR11	CHAR11	80
CH1	08	SEABUSE	CHAR11	CHAR11	80
CH2	08	ONE	REAL	REAL	80
ICONT	08	NEX	INTEGER	INTEGER	80
IDATA	08	NEX	INTEGER	INTEGER	80
IFADD	08	LEVEL	INTEGER	INTEGER	80
IFIND	08	LEVEL	INTEGER	INTEGER	80
INAPH	08	LEVEL	INTEGER	INTEGER	80
ISAUD	08	LEVEL	INTEGER	INTEGER	80
ISIN	08	LEVEL	INTEGER	INTEGER	80
ITOTL	08	LEVEL	INTEGER	INTEGER	80
LEVEL	08	LEVEL	LEVEL	LEVEL	80
LVL	08	LEVEL	LEVEL	LEVEL	80
NATT	08	LEVEL	LEVEL	LEVEL	80
NDEEP	08	LEVEL	LEVEL	LEVEL	80
NDIFF	08	LEVEL	LEVEL	LEVEL	80
NLOUD	08	LEVEL	LEVEL	LEVEL	80
NLUIS	08	LEVEL	LEVEL	LEVEL	80
NMODES	08	LEVEL	LEVEL	LEVEL	80
MSYS	08	LEVEL	LEVEL	LEVEL	80
NTAPE	08	LEVEL	LEVEL	LEVEL	80
SENS	08	LEVEL	LEVEL	LEVEL	80
SEMI	08	LEVEL	LEVEL	LEVEL	80
SYSENM	08	LEVEL	LEVEL	LEVEL	80
USER	08	LEVEL	LEVEL	LEVEL	80

```

--PROCEDURES--(LO-A)
--NAME--TYPE--CLASS--

```

NAME	TYPE	CLASS
COMPUT	SUBROUTINE	SUBROUTINE
DETNOD	SUBROUTINE	SUBROUTINE
GRAPH	SUBROUTINE	SUBROUTINE
PRETOT	SUBROUTINE	SUBROUTINE
SYSEN	SUBROUTINE	SUBROUTINE
TABDIS	SUBROUTINE	SUBROUTINE

[illegible]

SUBROUTINE COMPUT

```

68     INDX=INDXST(MCOUNT)
69     ENDIF
70     IF(CTRL.EQ.3) THEN
71         SUM=0.0
72         IFINDT=IRAY(IFIND,4)
73         DO 7 I=1,10
74             IF(IRAY(IFINDT,2).EQ.1) THEN
75                 GO TO 3
76             ENDIF
77             IFINDT=IRAY(IFINDT,4)
78             CONTINUE
79             IFINDT=IRAY(IFINDT,2)
80             DO 10 I=1,NCROSS
81                 IF(INDX.NE.IRAY(IFINDT,1)) SUM=SUM+IRAY(IFINDT,1)
82                 IFINDT=IRAY(IFINDT,3)
83                 CONTINUE
84             ENDIF
85             DO 40 I=0.20
86                 X=FLOAT(I)
87                 X=X*PI*180/3.14159
88                 X=X*PI*180/3.14159
89                 IF(SENSE.EQ.'SYST') THEN
90                     K=NCOUNT
91                     IFIND=IFINDT(MCOUNT)
92                     NCROSS=KIDSET(MCOUNT)
93                     INDX=INDXST(MCOUNT)
94                     IF(CTRL.EQ.3) THEN
95                         SUM=0.0
96                         IFINDT=IRAY(IFIND,4)
97                         DO 66 I=1,10
98                             IF(IRAY(IFINDT,2).EQ.1) THEN
99                                 GO TO 2
100                             ENDIF
101                             IFINDT=IRAY(IFINDT,4)
102                             CONTINUE
103                             IFINDT=IRAY(IFINDT,2)
104                             DO 12 I=1,NCROSS
105                                 IF(INDX.NE.IRAY(IFINDT,1)) SUM=SUM+IRAY(IFINDT,1)
106                                 IFINDT=IRAY(IFINDT,3)
107                                 CONTINUE
108                             ENDIF
109                             IF(CTRL.EQ.1) THEN
110                                 WHOLD(I,K)=(100.0*IRAY(IFIND,1SYS)+X)/(1.0-X)*((100.0
111                                     *IRAY(2,1SYS)
112                                     *-(100.0*IRAY(IFIND,1SYS)+IRAY(IFIND,2)))/(1.0-IRAY(IFIND,2)))
113                                 ELSEIF(CTRL.EQ.2) THEN
114                                     ATTHM=LABEL(IFIND)
115                                     TEMP=X*100.0
116                                     WHOLD(I,K)=(100.0*IRAY(2,1SYS)+IRAY(IFIND,2)
117                                         *X*100.0*ALU(TEMP,ATTHM))-100.0*IRAY(IFIND,1SYS)
118                                 ELSE
119                                     IFINDT=IRAY(IFIND,4)
120                                     DO 9 I=1,10
121                                         IF(IRAY(IFINDT,2).EQ.1) THEN
122                                             GO TO 6
123                                         ENDIF
124                                         IFINDT=IRAY(IFINDT,4)

```

SUBROUTINE COMPLY

```

115 CONTINUE
116 UNHOLD(I,K)=100.0*URAY(2,ISYS)-100.0*URAY(IFINDT,ISYS)
117 * XARRAY(IFINDT,2)
118 DO 60 L=1,NCROSS
119 IF(INDX.NE.L)THEN
120 UNHOLD(I,K)=UNHOLD(I,K)*(1.0-X)*100.0*URAY(IFINDT+L,ISYS)*
121 * XARRAY(IFINDT+L,1)/SUMXARRAY(IFINDT,2)
122 ELSE
123 UNHOLD(I,K)=UNHOLD(I,K)+X*100.0*URAY(IFINDT+L,ISYS)*XARRAY(IFINDT,2)
124 ENDIF
125 CONTINUE
126 NCOUNT=NCOUNT+1
127 IF(NCOUNT.LE.NSET)GO TO 11
128 ELSE
129 DO 30 K=1,NEVS
130 IF(CTRL.EG.1)THEN
131 UNHOLD(I,K)=100.0*URAY(IFIND,K)*X*((1.0-X)*X*(100.0*URAY(2,K)
132 * -100.0*URAY(IFIND,K)*XARRAY(IFIND,2))/(1.0-ARRAY(IFIND,2)))
133 ELSEIF(CTRL.EG.2)THEN
134 IF(K.EQ.ISYS)THEN
135 UNHOLD(I,K)=100.0*URAY(2,K)+XARRAY(IFIND,2)
136 ELSE
137 UNHOLD(I,K)=100.0*URAY(2,K)+X*100.0*URAY(IFIND,K)
138 UNHOLD(I,K)=100.0*URAY(2,K)
139 ENDIF
140 ELSE
141 IF(INDT=IRAY(IFIND,4)
142 DO 8 JM=1,12
143 IF(IRAY(IFINDT,2).EQ.IFINDT-1)THEN
144 GO TO 4
145 ENDIF
146 IF(INDT=IRAY(IFINDT,4)
147 CONTINUE
148 UNHOLD(I,K)=100.0*URAY(2,K)-100.0*URAY(IFINDT,K)*XARRAY(IFINDT,2)
149 DO 23 L=1,NCROSS
150 IF(INDX.NE.L)THEN
151 UNHOLD(I,K)=UNHOLD(I,K)*(1.0-X)*100.0*URAY(IFINDT+L,K)*
152 * XARRAY(IFINDT+L,1)/SUMXARRAY(IFINDT,2)
153 ELSE
154 UNHOLD(I,K)=UNHOLD(I,K)+X*100.0*URAY(IFINDT+L,K)*XARRAY(IFINDT,2)
155 ENDIF
156 CONTINUE
157 ENDIF
158 NCOUNT=1
159 CONTINUE
160 PRINT*,
161 RETURN
162 END

```

SUBROUTINE COMPUT

--VARIABLE MAP--(LO-A)	--NAME--ADDRESS--BLOCK--	--PROPERTIES--	--SIZE--	--TYPE--	--NAME--ADDRESS--BLOCK--	--PROPERTIES--	--TYPE--	--SIZE--
ATTACH	0B	/SEN1/		CHAR10	LVL	30B	/LEVEL/	INTEGER
CONTROL	0B	/SSAUSE/ SAU		INTEGER	MCOUNT	0B	/SEN4/	CHAR15
FINDST	0B	/SENS/	50	INTEGER	MCROSS	5B	/SSAUSE/ SAU	INTEGER
IFADD	4B	/SSAUSE/ SAU		INTEGER	MDIFF	2B	/SEN3/	INTEGER
IFIND	27B	/LEVEL/		INTEGER	MDUL	2B	/LEVEL/	INTEGER
IFINDT	25B	/LEVEL/		INTEGER	MSCT	52B	/LEVEL/	INTEGER
IT	101	/SSAUSE/ SAU		INTEGER	MSVS	0B	/SENS/	INTEGER
IT	13B	/SSAUSE/ SAU		INTEGER	SENS	0B	/SEN1/	CHAR15
IT	15B	/SSAUSE/ SAU		INTEGER	SUM	7B	/SSAUSE/ SAU	REAL
INDX	3B	/SEN2/		INTEGER	SVSNAM	1B	/SEN1/	CHAR10
INDXGT	63B	/SENS/	50	INTEGER	TEMP	14B	/SSAUSE/ SAU	REAL
INMRN	4B	/SEN2/	20	INTEGER	TEST	1B	/SSAUSE/ SAU	CHAR15
ISYS	5B	/SSAUSE/ SAU		INTEGER	UDLTA	3B	/SSAUSE/ SAU	REAL
J	12B	/SSAUSE/ SAU		INTEGER	UDLJ	0B	/SEN3/	REAL
KIDSLT	14B	/SENS/	50	INTEGER	UDLX	1B	/SEN2/	REAL
L	16B	/SSAUSE/ SAU		INTEGER	UDIN	6B	/SENS/	REAL
LEVEL	31B	/LEVEL/	60	INTEGER	UDIN	11B	/SSAUSE/ SAU	REAL

1701

--PROCEDURES--(LO-A)

--NAME--	--TYPE--	--ARGS--	--CLASS--	--NAME--	--TYPE--	--ARGS--	--CLASS--
ARRAY	REAL	2	FUNCTION	LABEL	CHAR10	1	FUNCTION
FLOAT	REAL	1	INTRINSIC	UDLU	REAL	2	FUNCTION
IRAY	INTEGER	2	FUNCTION	UDRY	REAL	2	FUNCTION

--STATEMENT LABELS--(LO-A)

--LABEL--ADDRESS--	--PROPERTIES--	--DEF	--LABEL--ADDRESS--	--PROPERTIES--	--DEF
2	254B	93	8	INACTIVE	14B
3	144B	69	9	DO-TERM	115
4	67B	149	10	DO-TERM	73
5	INACTIVE	39	11	DO-TERM	89
6	423B	116	12	DO-TERM	97
7	INACTIVE	68			

--ENTRY POINTS--(LO-A)

--NAME--ADDRESS--ARGS--

COMPUT 5B 0

--STATISTICS--

PROGRAM-UNIT LENGTH 12172 * 655
 CM LABELLED COMMON LENGTH 35523 * 1962
 CM STORAGE USED 62698 * 25384
 COMPILE TIME 0.382 SECONDS

```

1  SUBROUTINE DETNOD
2
3  C
4  C*****
5  C THIS ROUTINE ELICITS THAT NODE AT WHICH A DIRECT
6  C LSA, CSA, OR RSA IS TO BE PERFORMED. IT ALSO
7  C INITIALIZES THOSE VARIABLES REQUIRED BY COMPUT.
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

C*****
 THIS ROUTINE ELICITS THAT NODE AT WHICH A DIRECT
 LSA, CSA, OR RSA IS TO BE PERFORMED. IT ALSO
 INITIALIZES THOSE VARIABLES REQUIRED BY COMPUT.
 C*****
 CALLED BY: SENSTU
 C*****
 VARIABLES
 USED: IDATA, ICONT, IFIND, NATT, OBJECTU, SENS
 MODIFIED: ATTNAM, CH, I, IFINDT, II, INDEX, K,
 NCROSS, UNAX, UNAXI, UMIN, UMINI
 C*****
 COMMON/ATTR/NATT
 COMMON/LEVEL/LEVELS, INPRM(20), IFIND, NDIFF, IFADD, LUL, LEVEL(20,3)
 COMMON/NEC/ICONT, IDATA, I*TOTL
 COMMON/SEN1/ATTNAM, SENS, SYSNAM
 COMMON/SEN2/UMIN, UMAX, NCROSS, INDY, ISYS
 SAVE
 CHARACTER*10 SENS25, CH\$1, OBJECTU(16), ATT1, LABEL, SYSNAM, ATTNAM
 INDEX=0
 WRITE(*, '(5(/), 1X)')
 PRINT*, 'SENSITIVITY ANALYSIS FOLLOWS...'
 PRINT*, 'ENTER NNN FOR WHICH ', SENS, ' IS'
 PRINT*, 'TO BE PERTURBED...'
 CALL PREMFX
 IF (ICONT.EQ.0) THEN
 PRINT*, 'MODE DOES NOT EXIST.'
 ELSE
 IF (SENS.EQ.'LEVEL') AND (IDATA.NE.1) THEN
 PRINT*, 'A LEVEL ANALYSIS MAY ONLY BE PERFORMED
 PRINT*, 'ON A DATA NODE.'
 ICONT=0
 ENDIF
 IF (ICONT.EQ.0) THEN
 PRINT*, '(PRESS ANY LETTER TO CONTINUE)'
 READ(*, '(A1)') CH
 ENDIF
 IF (ICONT.NE.1) GO TO 1
 PRINT*,
 CALL OBJECT(IFIND, OBJECT)
 II=1
 DO 10 I=1, 4
 WRITE(*, '(1X, 4A10)') (OBJECTU(K), K=II, II+3)

SUBROUTINE DETNOD

```

58 11=11+4
59 CONTINUE
60 IF (SENS.EQ.'CURNT') THEN
61 PRINT*, 'CURRENT CUMLT IS ', ARAY(IFIND,2)
62 ELSEIF (SENS.EQ.'LEVEL') THEN
63 PRINT*, 'CURRENT RELUT IS ', ARAY(IFIND,1)
64 CALL SEMUAL
65 ELSE
66 PRINT*, '
67 PRINT*, 'CURRENT RELUT IS ', ARAY(IFIND,1)
68 ENDIF
69 INDEX=IRAY(IFIND,1)
70 NCROSS=0
71 IFINDT=IRAY(IFIND,4)
72 DO 5 I=1,10
73 IF (IRAY(IFINDT,2).EQ.IFINDT+1) THEN
74 GO TO 6
75 ENDOF
76 IFINDT=IRAY(IFINDT,4)
77 CONTINUE
78 IFINDT=IRAY(IFINDT,2)
79 DO 20 I=1,10
80 IF (IFINDT.GT.0) THEN
81 NCROSS=NCROSS+1
82 IFINDT=IRAY(IFINDT,3)
83 ELSE
84 GO TO 4
85 ENDOF
86 CONTINUE
87 IF (SENS.EQ.'LEVEL') THEN
88 DO 30 I=1,NATT
89 IF (ATT(I).EQ.LABEL(IFIND)) THEN
90 UNMAX=UNMAX1(ATT(I),ATT(I,2))
91 UNMIN=MIN1(ATT(I,2),ATT(I,2))
92 PRINT*, 'WE ARE WORKING WITH ATTRIBUTE: ',ATTNAM
93 ENDIF
94 CONTINUE
95 ELSE
96 UNMAX=1.0
97 UNMIN=0.0
98 ENDOF
99 PRINT*, '
100 PRINT*, 'MINIMUM 'SENS,' ('UNMIN,'-',UNMAX,') IS?'
101 READ(8,'(F5.2)') UNMIN1
102 IF ((UNMIN1.LT.UNMIN).OR.((UNMIN1.GT.1.0).AND.(SENS.NE.'LEVEL'))
103 & .OR.((UNMIN1.GT.UNMAX).AND.(SENS.EQ.'LEVEL')) ) GO TO 2
104 UNMIN=UNMIN1
105 PRINT*, '
106 PRINT*, 'MAXIMUM 'SENS,' ('UNMIN,'-',UNMAX,') IS?'
107 READ(8,'(F5.2)') UNMAX1
108 IF ((UNMAX1.LT.UNMIN).OR.((UNMAX1.GT.1.0).AND.(SENS.NE.'LEVEL'))
109 & .OR.((UNMAX1.GT.UNMAX).AND.(SENS.EQ.'LEVEL')) ) GO TO 3
110 UNMAX=UNMAX1
111 RETURN
112 END
113

```

SUBROUTINE DETMOD

--VARIABLE MAP--(LO-A)		--NAME--ADDRESS--BLOCK--		--PROPERTIES--		--TYPE--		--SIZE--	
ATTNAR	08	/SEMI/		CHAR#10					
CH	16	/SSASUSE/	SAU	INTEGER					60
DATA	248	/SSASUSE/	SAU	INTEGER					
IFADD	08	/NEA/		INTEGER					
IFINT	16	/NEA/		INTEGER					
IFINT	278	/LEVEL/		INTEGER					
IFINT	288	/LEVEL/		INTEGER					
IFINT	298	/SSASUSE/	SAU	INTEGER					16
INDEX	208	/SSASUSE/	SAU	INTEGER					
INDEX	38	/SEMI/		INTEGER					
INDEX	18	/LEVEL/		INTEGER					
ISYS	48	/SEMI/		INTEGER					
ISYS	28	/NEA/		INTEGER					
ISYS	248	/SSASUSE/	SA	INTEGER					

--PROCEDURES--(LO-A)

--NAME--		--TYPE--		--CLASS--		--ARGS--		--TYPE--		--NAME--ADDRESS--BLOCK--		--PROPERTIES--		--TYPE--		--SIZE--	
MAXI	REAL			INTRINSIC		7		IRAY	INTEGER								
MINI	REAL			INTRINSIC		7		IRAY	INTEGER								
MAXI	REAL			FUNCTION		2		IRAY	INTEGER								
ATT	CHAR#10			FUNCTION		2		IRAY	INTEGER								
ATT	CHAR#10			FUNCTION		2		IRAY	INTEGER								

--STATEMENT LABELS--(LO-A)

--NAME--ADDRESS--BLOCK--		--PROPERTIES--		--DEF--		--NAME--ADDRESS--BLOCK--		--PROPERTIES--		--DEF--	
1	118			33		4	2452			67	
2	2818			100		5	INACTIVE			74	
3	3518			106		6	2168			74	

--ENTRY POINTS--(LO-A)

--NAME--ADDRESS--BLOCK--		--PROPERTIES--		--DEF--	
DETRON	58			0	

--STATISTICS--

PROGRAM UNIT LENGTH 7109 - 456
 CM LABELLED COMMON LENGTH 1718 - 121
 CM STORAGE USED 62508 - 25384
 COMPILE TIME 0.231 SECONDS

```

1  SUBROUTINE GNDRI(1,CH)
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

THIS ROUTINE GENERATES THE X-AXIS LABELS FOR THE
 GRAPHICAL DISPLAY OF SENSITIVITY RESULTS.

CALLED BY: GRAPH

VARIABLES
 USED: CH, I, NSET, NSVS, SENS, WHOLD
 MODIFIED: IDLTA, IDUM, I1, K, MAX, MIN, MM, UMAX, UMIN

COMMON/SENS/ATTHAY,SENS,SVSHAM
 COMMON/SENS/UMIN,UMAX,NCROSS,INDX,ISVS
 COMMON/SENS/WHOLD(1:20,0:60)
 COMMON/SENS/FINDST(50),NSET,INDXST(50),KIDSET(50)
 COMMON/SENS/NSVS
 INTEGER IDUM(15)
 CHARACTER CH,ATTHAY(10),SENS(5,5),CROSS(10)
 UMIN=0.0
 UMAX=0.0
 IF(CH.EQ.'E')THEN
 IF(SENS.NE.'SYSTM')THEN
 MM=NSVS
 ELSE
 MM=NSET
 ENDIF
 DO 10 I=1,20
 DO 10 K=1,MM
 IF(UMIN.GT.WHOLD(I1,K))UMIN=WHOLD(I1,K)
 IF(UMAX.LT.WHOLD(I1,K))UMAX=WHOLD(I1,K)
 CONTINUE
 ELSE
 UMIN=0.0
 UMAX=0.0
 ENDIF
 MIN=INT(UMIN/20.0)*20
 MAX=INT(UMAX/10.0)*20.0/20
 IDLTA=(MAX-MIN)/5
 IF(IDLTA.LT.1)THEN
 PRINT*,
 DO 20 I=0,5
 IDUM(I)=MIN+IDLTA*I
 WRITE(1, '(2X,5X,15)')((IDUM(I),I=0,5)
 ELSE
 DO 30 I=0,5

OVERALL VALUE

SUBROUTINE CHDR

```

50 IDUM(11)=MIN+IDLTA11
50 WRITE(2, '(2X,G15X,15)') (IDUM(11),11=0,5)
60 PRINT*,
61 ENDIF
62 RETURN
63 END

```

OVERALL VALUE

--VARIABLE MAP--(LO-A)			--NAME--ADDRESS--BLOCK--PROPERTIES--TYPE--SIZE		
ATTNAM	02	/SENS/	MAX	2738	INTEGER
CH	2	DUNNY-ARG	FIN	2728	INTEGER
FINDST	08	/SENS/	NA	2658	INTEGER
I	1	DUNNY-ARG	NCROSS	28	/SENS/
IDLTA	2748		NSET	628	INTEGER
IDUN	2578		NSYS	08	INTEGER
II	2568		SENS	18	CHAR*10
INDX	38	/SENS2/	SYSHAM	18	REAL
INDXST	632	/SENS/	WHOLD	08	REAL
ISVS	48	/SENS2/	UMAX	18	REAL
K	2708		UMIN	08	REAL
KIDSET	1428	/SENS/			

--PROCEDURES--(LO-A)

INT GENEPIC) INTRINSIC

--STATEMENT LABELS--(LO-A)

--LABEL-ADDRESS--PROPERTIES--DEF

--ENTRY POINTS--(LO-A)

--NAME--ADDRESS--ARGS--

--STATISTICS--

PROGRAM-UNIT LENGTH 3018 183
CM LABELLED COMMON LENGTH 35058 1861
CM STORAGE USED 53608 24860
COMPILE TIME 6.100 SECONDS

[illegible]

SUBROUTINE GRAPH

```

50 DO 10 JJ=0,1
51 PRINT*, 'SYMBOL-', SYMBOL(I1+JJ), ' IS NODE:'
52 IN=1
53 CALL OBJECT(FINDST(I1+JJ), OBJECTU)
54 DO 6 JA=1,4
55 WRITE(X, '(IX,4A10)')(OBJECTU(M), M=1M, IN=3)
56 IN=IN+4
57 CONTINUE
58 PRINT*, '
59 CONTINUE
60 ENDF
61
62 I1=I1+2
63 CONTINUE
64 IF (SENS.NE. 'SYSTEM') THEN
65 IF (INDEX$2,1, LIMIT) PRINT*, 'SYMBOL-', SYMBOL(LIMIT),
66 ' IS SYSTEM', 'SYSLBL(LIMIT)
67 ELSE
68 IF (INDEX$2,1,1, LIMIT) THEN
69 IN=1
70 CALL OBJECT(FINDST(LIMIT), OBJECTU)
71 PRINT*, 'SYMBOL-', SYMBOL(LIMIT), ' IS NODE:'
72 DO 7 JA=1,4
73 WRITE(X, '(IX,4A10)')(OBJECTU(M), M=1M, IN=3)
74 IN=IN+4
75 CONTINUE
76 ENDF
77
78 IF (SENS.NE. 'LEVEL') THEN
79 PRINT*, 'SENSITIVITY ANALYSIS ON', SENS
80 IF (SENS.EQ. 'SYSTEM') PRINT*, 'FOR SYSTEM', SYSLBL(1SYS)
81 ELSE
82 PRINT*, 'LEVEL ANALYSIS ON SYSTEM', SYSNOH
83 PRINT*, 'FOR ATTRIBUTE', ATTRNOH
84 ENDF
85 CALL GHDR(1, CH)
86 DO 20 I=1,20
87 DO 30 J=1,50
88 IF (I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.
89 = (I.EQ.20)) THEN
90 LINE(J)=1
91 ELSE
92 LINE(J)=0
93 LINE(J)=0
94 ENDF
95 DO 40 J=1,51,10
96 CONTINUE
97 DO 50 J=1, LIMIT
98 KVAL=(UMOLD(I,J)-UMIN)/(UMAX-UMIN)/50.0
99 INPUT=INT(KVAL+.75*(UMAX-UMIN)/50.0)
100 IF (INPUT.LE.0) INPUT=1
101 IF (INPUT.GT.51) INPUT=51
102 LINE(OUTPUT)=SYMBOL(J)
103 CONTINUE
104 IF ((I.EQ.1).OR.(I.EQ.5).OR.(I.EQ.10).OR.(I.EQ.15).OR.
105 = (I.EQ.20)) THEN
106 WRITE(X, '(IX,F10.2,51A1)') UMOLD(I,0), (LINE(X), X=1,51)
107
108
109
110
111
112
113
114

```

SUBROUTINE GRAPH

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARG--
GRAPH 58 0

--STATISTICS--
PROGRAM-UNIT LENGTH 10538 * 555
CM LABELLED COMMON LENGTH 35578 * 1503
CM STORAGE USED 625008 * 25984
CM STORAGE USED 0.306 SECONDS
COMPILE TIME

```

SUBROUTINE GRAPH

```

115 ELSE
116 PRINT,
117 ENDIF
118 CONTINUE
119 CALL ICHEP(2,CH)
120 PRINT, '(PRESS ANY LETTER TO CONTINUE)'
121 PRINT, '(A1)'
122 READ(X, '(A1)')
123 RETURN
124 END
125

```

NAME	ADDRESS	BLOCK	PR	PROPERTIES	TYPE	SIZE
ATTNAM	0B	/SENS/	SAU		CHAR10	
CH	2B	/SENS/	SAU		CHAR10	
FINDST	0B	/SENS/	SAU		INTEGER	50
I	41B	/SENS/	SAU		INTEGER	
I	37B	/SENS/	SAU		INTEGER	
I	43B	/SENS/	SAU		INTEGER	
IN	40B	/SENS/	SAU		INTEGER	
INDEX	3B	/SENS/	SAU		INTEGER	50
INDX	63B	/SENS/	SAU		INTEGER	
INDXST	50B	/SENS/	SAU		INTEGER	
IPUT	4B	/SENS/	SAU		INTEGER	
ISVS	46B	/SENS/	SAU		INTEGER	
J	42B	/SENS/	SAU		INTEGER	
JJ	44B	/SENS/	SAU		INTEGER	
JN	51B	/SENS/	SAU		INTEGER	
KIDSET	145B	/SENS/	SAU		INTEGER	50

PROCEDURES--(LO-A)

NAME	TYPE	ARG	CLASS
CMR	GENERIC	2	SUBROUTINE
INT	INTEGER	1	INTRINSIC
MINO	INTEGER	2	INTRINSIC
OBJECT	CHAR10	1	SUBROUTINE
SVSLBL	CHAR10	1	FUNCTION

STATEMENT LABELS--(LO-A)

NAME	ADDRESS	PROPERTIES	DEF
1	10B	DO-TERM	28
2	INACTIVE	DO-TERM	58
3	INACTIVE	DO-TERM	56

STATEMENT LABELS--(LO-A)

NAME	ADDRESS	PROPERTIES	DEF
39	INACTIVE	DO-TERM	102
40	INACTIVE	DO-TERM	104
50	INACTIVE	DO-TERM	111

[illegible]

```

      END
      46
      VARIABLE MAP--(LO-n)
      NAME-----ADDRESS-BLOCK-----PROPERTIES-----TYPE-----SIZE
      ANSWER          3    DIGNY-ARG        CHAR879
      ATTNAME         08     /SEM1/         CHAR316
      I               1   DIGNY-ARG        INTEGER
      IFADD           27B    /LEVEL/       INTEGER
      IFIND            25B   /LEVEL/       INTEGER
      INDX             3B    /SEM2/       INTEGER
```

SUBROUTINE HEADER 7474 OPT=0
 --NAME-- ADDRESS--BLOCK-- PROPERTIES--
 --NAME-- ADDRESS--BLOCK-- PROPERTIES-- TYPE-- SIZE
 NAME 5.1-520 11/23/81 19.35.01 2 6122
 NAME-- ADDRESS--BLOCK-- PROPERTIES-- TYPE-- SIZE
 NAME 5.1-520 11/23/81 19.35.01 2 6122

--PROCEDURES-- (LO-A)
 --NAME-- TYPE-- ARGS-- CLASS--
 OBJECT 2 SUBROUTINE

--ENTRY POINTS-- (LO-A)
 --NAME-- ADDRESS-- ARGS--
 HEADER 58 3

--STATISTICS--
 PROGRAM-UNIT LENGTH : 638 - 115
 CR LABELLED COMMON LENGTH : 468 - 96
 CR STORAGE USED 606808 - 24960
 COMPILE TIME 0.065 SECONDS

```

18 SUBROUTINE MODSET
19 C
20 C*****
21 C THIS ROUTINE ELICITS THE SET OF NODES OVER WHICH
22 C THE SSA IS TO BE PERFORMED (ANALOGOUS TO DETMOD).
23 C
24 C CALLED BY: SYSEN
25 C
26 C VARIABLES
27 C   USED:  IDATA,IFIND,NTYPE,ORJCTV
28 C   MODIFIED:  CH,FINDET,ICONT,1,INDXST,K,
29 C             NCOUNT,MSET,UMAX,UMIN,UMINI
30 C*****
31 C
32 C COMMON/LEVEL/ NLULS,IMRN(20),IFIND,NDIFF,IFADD,LUL,LEVEL(20,3)
33 C COMMON/NEP/ ICONT, IDATA, ITOTL
34 C COMMON/SEN1/ ATTNAM,SEMS,SYSNAM
35 C COMMON/SEN2/ UNIK,UMAX,NCROSS,INDX,ISYS
36 C COMMON/SEN4/ NTYPE
37 C COMMON/SEMS/ FINDS(50),MSET,INDXST(50),KIDSET(50)
38 C SAVE
39 C CHARACTER*10 SEMSIS,CH*1,OBJCTU(16),ATT1,LABEL,SYSNAM,ATTNAM,
40 C   * RTYPEIS
41 C INTEGER FINDST,INXST,KIDSET
42 C WRITE(3, '(5(/),1X)',)
43 C PRINT*, 'HOW MANY NODES WOULD YOU LIKE TO EXAMINE? (1-50)'
44 C PRINT*, ' '
45 C READ(3, '(12)',MSET)
46 C IF((MSET.LT.1).OR.(MSET.GT.50))THEN
47 C   PRINT*, 'INPUT VALUE OUT OF RANGE'
48 C   GO TO 1
49 C ENDIF
50 C NCOUNT=0
51 C WRITE(3, '(5(/),1X)',)
52 C INDXST(NCOUNT+1)=0
53 C PRINT*, 'ENTER NIN FOR WHICH 'NTYPE' IS
54 C PRINT*, 'TO BE PERTURBED...'
55 C CALL PERNEX
56 C IF ICONT.EQ.0 THEN
57 C   PRINT*, 'NODE DOES NOT EXIST'
58 C ELSE
59 C   IF (NTYPE.EQ. 'VALUE') .AND. (IDATA.NE.1) THEN
60 C     PRINT*, 'A VALUE ANALYSIS MAY ONLY BE PERFORMED'
61 C     PRINT*, 'ON A DATA NODE.'
62 C     ICONT=0
63 C   ENDIF
64 C   IF (ICONT.EQ.0) THEN
65 C     PRINT*, '(PRESS ANY LETTER TO CONTINUE)'
66 C
67 C

```

SUBROUTINE NOSSET

```

58 PRINT, '(A1)'CH
59 READ(2, '(A1)'CH)
60 ENDIF
61 IF (ICOUNT.NE.1) GO TO 2
62 NCOUNT=NCOUNT+1
63 FINDSET(NCOUNT)=IFIND
64 PRINT, '
65 CALL OBJECT(IFIND,OBJECT)
66 PRINT, 'THE OBJECTIVE IS'
67 IF 1=1
68 DO 10 1=1,4
69 WRITE(1, '(1X,4F10.1)') (OBJECTU(K),K=1,11,1+3)
70 IF 1=1+4
71 CONTINUE
72 INDEXST(NCOUNT)=IRAY(IFIND,1)
73 KIDSET(NCOUNT)=0
74 IFINDT=IRAY(IFIND,4)
75 DO 20 1=1,10
76 IF (IRAY(IFINDT,2).EQ.IFINDT+1) GO TO 3
77 IFINDY=IRAY(IFINDT,4)
78 CONTINUE
79 IFINDY=IRAY(IFINDT,2)
80 DO 30 1=1,10
81 IF (IFINDT.GT.0) THEN
82 KIDSET(NCOUNT)=KIDSET(NCOUNT)+1
83 IFINDY=IRAY(IFINDT,3)
84 ELSE
85 GO TO 4
86 ENDIF
87 CONTINUE
88 IF (NCOUNT.LT.NSET) GO TO 2
89 UMIN=0.0
90 UMAX=1.0
91 IF (TYPE.EQ.'VALUE') UMAX=100.0
92 PRINT, 'MINIMUM', NTYPE, '(', UMIN, '-', UMAX, ') IS?'
93 READ(2, '(F5.2)') UMINI
94 IF ((UMIN.LT.0.0).OR.(UMINI.GT.UMAX)) THEN
95 PRINT, 'INPUT IS OUT OF RANGE'
96 GO TO 5
97 ENDIF
98 UMIN=UMINI
99 PRINT, 'MAXIMUM', NTYPE, '(', UMIN, '-', UMAX, ') IS?'
100 READ(2, '(F5.2)') UMAXI
101 IF ((UMAXI.LT.UMIN).OR.(UMAXI.GT.UMAX)) THEN
102 PRINT, 'INPUT IS OUT OF RANGE'
103 GO TO 6
104 ENDIF
105 UMAX=UMAXI
106 RETURN
107 END
108
109
110

```

[illegible]

NAME	TYPE	APCS	FUNCTION SUBROUTINE SUBROUTINE	CLASS
PROCEDURES--(LO-A)				
IRAY	INTEGER	2		
OBJECT		2		
PREFIX		0		

```

PREMEX
--STATEMENT LABELS--(L0-A)
-LABEL-ADDRESS-----PROPERTIES-----DEF
1       78      1
24      34      2
43      43      3
5       5       4
2418    93      5
2558    10      6
-LABEL-ADDRESS-----PROPERTIES-----DEF
10      10      7
10      10      8
10      10      9
10      10     10
10      10     11
10      10     12
10      10     13
10      10     14
10      10     15
10      10     16
10      10     17
10      10     18
10      10     19
10      10     20
10      10     21
10      10     22
10      10     23
10      10     24
10      10     25
10      10     26
10      10     27
10      10     28
10      10     29
10      10     30
10      10     31
10      10     32
10      10     33
10      10     34
10      10     35
10      10     36
10      10     37
10      10     38
10      10     39
10      10     40
10      10     41
10      10     42
10      10     43
10      10     44
10      10     45
10      10     46
10      10     47
10      10     48
10      10     49
10      10     50
10      10     51
10      10     52
10      10     53
10      10     54
10      10     55
10      10     56
10      10     57
10      10     58
10      10     59
10      10     60
10      10     61
10      10     62
10      10     63
10      10     64
10      10     65
10      10     66
10      10     67
10      10     68
10      10     69
10      10     70
10      10     71
10      10     72
10      10     73
10      10     74
10      10     75
10      10     76
10      10     77
10      10     78
10      10     79
10      10     80
10      10     81
10      10     82
10      10     83
10      10     84
10      10     85
10      10     86
10      10     87
10      10     88
10      10     89
10      10     90
10      10     91
10      10     92
10      10     93
10      10     94
10      10     95
10      10     96
10      10     97
10      10     98
10      10     99
10      10    100
10      10    101
10      10    102
10      10    103
10      10    104
10      10    105
10      10    106
10      10    107
10      10    108
10      10    109
10      10    110
10      10    111
10      10    112
10      10    113
10      10    114
10      10    115
10      10    116
10      10    117
10      10    118
10      10    119
10      10    120
10      10    121
10      10    122
10      10    123
10      10    124
10      10    125
10      10    126
10      10    127
10      10    128
10      10    129
10      10    130
10      10    131
10      10    132
10      10    133
10      10    134
10      10    135
10      10    136
10      10    137
10      10    138
10      10    139
10      10    140
10      10    141
10      10    142
10      10    143
10      10    144
10      10    145
10      10    146
10      10    147
10      10    148
10      10    149
10      10    150
10      10    151
10      10    152
10      10    153
10      10    154
10      10    155
10      10    156
10      10    157
10      10    158
10      10    159
10      10    160
10      10    161
10      10    162
10      10    163
10      10    164
10      10    165
10      10    166
10      10    167
10      10    168
10      10    169
10      10    170
10      10    171
10      10    172
10      10    173
10      10    174
10      10    175
10      10    176
10      10    177
10      10    178
10      10    179
10      10    180
10      10    181
10      10    182
10      10    183
10      10    184
10      10    185
10      10    186
10      10    187
10      10    188
10      10    189
10      10    190
10      10    191
10      10    192
10      10    193
10      10    194
10      10    195
10      10    196
10      10    197
10      10    198
10      10    199
10      10    200
10      10    201
10      10    202
10      10    203
10      10    204
10      10    205
10      10    206
10      10    207
10      10    208
10      10    209
10      10    210
10      10    211
10      10    212
10      10    213
10      10    214
10      10    215
10      10    216
10      10    217
10      10    218
10      10    219
10      10    220
10      10    221
10      10    222
10      10    223
10      10    224
10      10    225
10      10    226
10      10    227
10      10    228
10      10    229
10      10    230
10      10    231
10      10    232
10      10    233
10      10    234
10      10    235
10      10    236
10      10    237
10      10    238
10      10    239
10      10    240
10      10    241
10      10    242
10      10    243
10      10    244
10      10    245
10      10    246
10      10    247
10      10    248
10      10    249
10      10    250
10      10    251
10      10    252
10      10    253
10      10    254
10      10    255
10      10    256
10      10    257
10      10    258
10      10    259
10      10    260
10      10    261
10      10    262
10      10    263
10      10    264
10      10    265
10      10    266
10      10    267
10      10    268
10      10    269
10      10    270
10      10    271
10      10    272
10      10    273
10      10    274
10      10    275
10      10    276
10      10    277
10      10    278
10      10    279
10      10    280
10      10    281
10      10    282
10      10    283
10      10    284
10      10    285
10      10    286
10      10    287
10      10    288
10      10    289
10      10    290
10      10    291
10      10    292
10      10    293
10      10    294
10      10    295
10      10    296
10      10    297
10      10    298
10      10    299
10      10    300
10      10    301
10      10    302
10      10    303
10      10    304
10      10    305
10      10    306
10      10    307
10      10    308
10      10    309
10      10    310
10      10    311
10      10    312
10      10    313
10      10    314
10      10    315
10      10    316
10      10    317
10      10    318
10      10    319
10      10    320
10      10    321
10      10    322
10      10    323
10      10    324
10      10    325
10      10    326
10      10    327
10      10    328
10      10    329
10      10    330
10      10    331
10      10    332
10      10    333
10      10    334
10      10    335
10      10    336
10      10    337
10      10    338
10      10    339
10      10    340
10      10    341
10      10    342
10      10    343
10      10    344
10      10    345
10      10    346
10      10    347

```

```

--ENTRY POINTS--(LO-A)
NAME--ADDRESS--ARGS--

```

135004 25 9

--STATISTICS--

PROGRAM-UNIT LENGTH	5740	•	380
CR LABELLED COMMON LENGTH	4238	•	275
CM STORAGE USED	80600	•	2460
COMPILE TIME	9.209	SECONDS	

```

1 100 * 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
SUBROUTINE SENTVP
C*****
C1 THIS ROUTINE ESTABLISHES THE SUBTYPE OF SSA "O BE
C2 PERFORMED.
C3
C4 CALLED BY: SVSSEN
C5
C6 VARIABLES
C7 USED: (NONE)
C8
C9 MODIFIED: CH,MTYPE
C10
C11*****
C12
C13 COMMON/SEN4/MTYPE
C14
C15 SAVE
C16 CHARACTERS MTYPE,CH
C17
C18 PRINTX,'WHAT TYPE OF ANALYSIS WOULD YOU LIKE TO DO...'
C19 PRINTX,'C' INPUT
C20 PRINTX,'C' INPUT
C21 PRINTX,'C' INPUT
C22
C23 READX,'(C1)' CH
C24 IF ((CH.NE.'C').AND.(CH.NE.'R').AND.(CH.NE.'V')) THEN
C25 PRINTX,'PLEASE INPUT 'C' OR 'R' OR 'V'
C26 GO TO 1
C27
C28 ENDIF
C29 IF (CH.EQ.'C') THEN
C30 MTYPE='CURLY'
C31 ELSEIF (CH.EQ.'R') THEN
C32 MTYPE='RELT'
C33 ELSE
C34 MTYPE='VALUE'
C35 ENDIF
C36 RETURN
C37 END
C38
C39
C40
C41
C42
C43
C44
C45
C46
C47
C48
C49
C50
C51
C52
C53
C54
C55
C56
C57
C58
C59
C60
C61
C62
C63
C64
C65
C66
C67
C68
C69
C70
C71
C72
C73
C74
C75
C76
C77
C78
C79
C80
C81
C82
C83
C84
C85
C86
C87
C88
C89
C90
C91
C92
C93
C94
C95
C96
C97
C98
C99
C100
C101
C102
C103
C104
C105
C106
C107
C108
C109
C110
C111
C112
C113
C114
C115
C116
C117
C118
C119
C120
C121
C122
C123
C124
C125
C126
C127
C128
C129
C130
C131
C132
C133
C134
C135
C136
C137
C138
C139
C140
C141
C142
C143
C144
C145
C146
C147
C148
C149
C150
C151
C152
C153
C154
C155
C156
C157
C158
C159
C160
C161
C162
C163
C164
C165
C166
C167
C168
C169
C170
C171
C172
C173
C174
C175
C176
C177
C178
C179
C180
C181
C182
C183
C184
C185
C186
C187
C188
C189
C190
C191
C192
C193
C194
C195
C196
C197
C198
C199
C200
C201
C202
C203
C204
C205
C206
C207
C208
C209
C210
C211
C212
C213
C214
C215
C216
C217
C218
C219
C220
C221
C222
C223
C224
C225
C226
C227
C228
C229
C230
C231
C232
C233
C234
C235
C236
C237
C238
C239
C240
C241
C242
C243
C244
C245
C246
C247
C248
C249
C250
C251
C252
C253
C254
C255
C256
C257
C258
C259
C260
C261
C262
C263
C264
C265
C266
C267
C268
C269
C270
C271
C272
C273
C274
C275
C276
C277
C278
C279
C280
C281
C282
C283
C284
C285
C286
C287
C288
C289
C290
C291
C292
C293
C294
C295
C296
C297
C298
C299
C300
C301
C302
C303
C304
C305
C306
C307
C308
C309
C310
C311
C312
C313
C314
C315
C316
C317
C318
C319
C320
C321
C322
C323
C324
C325
C326
C327
C328
C329
C330
C331
C332
C333
C334
C335
C336
C337
C338
C339
C340
C341
C342
C343
C344
C345
C346
C347
C348
C349
C350
C351
C352
C353
C354
C355
C356
C357
C358
C359
C360
C361
C362
C363
C364
C365
C366
C367
C368
C369
C370
C371
C372
C373
C374
C375
C376
C377
C378
C379
C380
C381
C382
C383
C384
C385
C386
C387
C388
C389
C390
C391
C392
C393
C394
C395
C396
C397
C398
C399
C400
C401
C402
C403
C404
C405
C406
C407
C408
C409
C410
C411
C412
C413
C414
C415
C416
C417
C418
C419
C420
C421
C422
C423
C424
C425
C426
C427
C428
C429
C430
C431
C432
C433
C434
C435
C436
C437
C438
C439
C440
C441
C442
C443
C444
C445
C446
C447
C448
C449
C450
C451
C452
C453
C454
C455
C456
C457
C458
C459
C460
C461
C462
C463
C464
C465
C466
C467
C468
C469
C470
C471
C472
C473
C474
C475
C476
C477
C478
C479
C480
C481
C482
C483
C484
C485
C486
C487
C488
C489
C490
C491
C492
C493
C494
C495
C496
C497
C498
C499
C500
C501
C502
C503
C504
C505
C506
C507
C508
C509
C510
C511
C512
C513
C514
C515
C516
C517
C518
C519
C520
C521
C522
C523
C524
C525
C526
C527
C528
C529
C530
C531
C532
C533
C534
C535
C536
C537
C538
C539
C540
C541
C542
C543
C544
C545
C546
C547
C548
C549
C550
C551
C552
C553
C554
C555
C556
C557
C558
C559
C560
C561
C562
C563
C564
C565
C566
C567
C568
C569
C570
C571
C572
C573
C574
C575
C576
C577
C578
C579
C580
C581
C582
C583
C584
C585
C586
C587
C588
C589
C590
C591
C592
C593
C594
C595
C596
C597
C598
C599
C600
C601
C602
C603
C604
C605
C606
C607
C608
C609
C610
C611
C612
C613
C614
C615
C616
C617
C618
C619
C620
C621
C622
C623
C624
C625
C626
C627
C628
C629
C630
C631
C632
C633
C634
C635
C636
C637
C638
C639
C640
C641
C642
C643
C644
C645
C646
C647
C648
C649
C650
C651
C652
C653
C654
C655
C656
C657
C658
C659
C660
C661
C662
C663
C664
C665
C666
C667
C668
C669
C670
C671
C672
C673
C674
C675
C676
C677
C678
C679
C680
C681
C682
C683
C684
C685
C686
C687
C688
C689
C690
C691
C692
C693
C694
C695
C696
C697
C698
C699
C700
C701
C702
C703
C704
C705
C706
C707
C708
C709
C710
C711
C712
C713
C714
C715
C716
C717
C718
C719
C720
C721
C722
C723
C724
C725
C726
C727
C728
C729
C730
C731
C732
C733
C734
C735
C736
C737
C738
C739
C740
C741
C742
C743
C744
C745
C746
C747
C748
C749
C750
C751
C752
C753
C754
C755
C756
C757
C758
C759
C760
C761
C762
C763
C764
C765
C766
C767
C768
C769
C770
C771
C772
C773
C774
C775
C776
C777
C77
```

```

1  SUBROUTINE SENVAL
2
3  C
4  C
5  C
6  C
7  C
8  C
9  C
10 C
11 C
12 C
13 C
14 C
15 C
16 C
17 C
18 C
19 C
20 C
21 C
22 C
23 C
24 C
25 C
26 C
27 C
28 C
29 C
30 C
31 C
32 C
33 C
34 C
35 C
36 C
37 C
38 C
39 C
40 C
41 C
42 C
43 C
44 C
45 C
46 C
47 C
48 C
49 C
50 C
51 C
52 C
53 C
54 C
55 C
56 C
57 C

```

THIS ROUTINE ESTABLISHES THE SPECIFIC ALTERNATIVE SYSTEM
 TO BE EXAMINED FOR LSA AND SSA.

CALLED BY: DETMOD,SVSSEN

VARIABLES
 USED: ATTHAR,IFIND,MTYPE,NSVS,SENS
 MODIF:FD: ANSWER,CH,1,1,1,1,INDEX,ISVS,K,SYSNAM,TSYS

COMMON/LEVEL/HLULS,IMRN(20),IFIND,NDIFF,IFADD,LVL,LEVEL(20,3)
 COMMON/SEN1/ATTHAR,SENS,SYSNAM
 COMMON/SEN2/UMIN,UFAX,NCROSS,INDEX,ISVS
 COMMON/SVS1/NSVS
 SAVE

CHARACTER*10 SYSLBL,ANSWER,SYSNAM,SENSES,CH1,TSVS(4),ATTNAR
 WRITE(6, '(5(1),1X)',
 PRINT*, 'SYSTEMS AVAILABLE:'
 INDEX=NSVS/4
 II=1
 DO 10 I=1,INDEX
 JJ=1
 DO 9 K=II,II+3
 TSVS(IJ)=SYSLBL(K)
 IJ=IJ+1
 CONTINUE
 WRITE(6, '(1X,4(10,5X))')(TSVS(K),K=1,4)
 II=II+4
 CONTINUE
 IF(INDEX*4.LT.NSVS)THEN
 IJ=1
 DO 11 K=INDEX*4+1,NSVS
 TSVS(IJ)=SYSLBL(K)
 IJ=IJ+1
 CONTINUE
 WRITE(6, '(1X,4(10,5X))')(TSVS(K),K=1,IJ-1)
 ENDIF
 PRINT*,
 IF(SENSEN, 'SYSTEM')THEN
 PRINT*, 'ENTER SYSTEM OF WHICH 'ATTNAR' IS TO BE'
 PRINT*, 'PERTURBED...'
 ELSE
 PRINT*, 'WHAT SYSTEM IS TO BE ANALYZED?'
 ENDIF
 PRINT*, ' ?'

SUBROUTINE SENVAL

```

58 READL, '(A10)' ANSWER
59 ISYS=0
60 DO 20 I=1, NSYS
61 IF (ANSWER.CO.SVSLBL(I)) THEN
62   ISYS=I
63   SYSSAM=ANSWER
64   ENDIF
65 CONTINUE
66 IF (ISYS.EQ.0) THEN
67   PRINT, 'SYSTEM ENTERED IS NOT VALID.'
68   PRINT, 'PRESS ANY LETTER TO CONTINUE.'
69   GO TO 1
70   READL, '(A1)' CH
71   GO TO 1
72   ENDIF
73 PRINT, ' '
74 IF (SENS.NE. 'SYSTM') THEN
75   PRINT, 'CURRENT MODE LEVEL IS 'XLEVEL'(URAY(FIND,ISYS),ATTNAM)
76   ENDIF
77 RETURN
78 END

```

NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE	NAME	ADDRESS	BLOCK	PROPERTIES	TYPE	SIZE
ANSWER	12	12	SENSEUSE/ SAV	CHAR10		K	128	128	SENSEUSE/ SAV	INTEGER	60
ATTNAM	68	68	SEN1/	CHAR10		LEVEL	312	312	LEVEL/	INTEGER	
CM	28	28	SENSEUSE/ SAV	CHAR10		LUL	308	308	LUL/	INTEGER	
I	118	118	SENSEUSE/ SAV	INTEGER		NCROSS	258	258	SEN2/	INTEGER	
IF ADD	278	278	LEVEL/	INTEGER		NDIFF	258	258	LEVEL/	INTEGER	
IF IND	258	258	LEVEL/	INTEGER		NLULS	08	08	LEVEL/	INTEGER	
I	108	108	SENSEUSE/ SAV	INTEGER		NSYS	08	08	SEN1/	CHAR10	
IJ	128	128	SENSEUSE/ SAV	INTEGER		SENS	18	18	SEN1/	CHAR10	
INDEX	78	78	SENSEUSE/ SAV	INTEGER		TSYS	32	32	SEN2/	REAL	4
INDX	38	38	SEN2/	INTEGER	20	UNAX	08	08	SEN2/	REAL	
INRM	18	18	LEVEL/	INTEGER		UNIN	08	08	SEN2/	REAL	
ISYS	48	48	SEN2/	INTEGER							

PROCEDURES--(LO-A) SUBROUTINE SENVAL

NAME	TYPE	CLASS	FUNCTION	ENTRY POINTS--(LO-A)	NAME	ADDRESS	ARGS
SVSLBL	CHAR10	1	FUNCTION				
URAY	REAL	2	FUNCTION				
XLEVEL	REAL	2	FUNCTION				

STATEMENT LABELS--(LO-A)

NAME	ADDRESS	DEF	PROPERTY	DEF
1	78	28	DO-TERM	28
10	INACTIVE	41	DO-TERM	41
11	INACTIVE	47	DO-TERM	47
20	INACTIVE	65	DO-TERM	65

PROGRAM-UNIT LENGTH
CM LABELLED COMMON LENGTH
CM STORAGE USED
COMPILE TIME

4448 * 208
1528 * 108
59508 * 24960
0.144 SECONDS

```

1  SUBROUTINE SVSSEN
2
3  C
4  C*****
5  C THIS ROUTINE INITIALIZES THE PROGRAM FOR CONDUCTING
6  C SSA.
7  C
8  C
9  C
10 C
11 C CALLED BY: SEMSTV
12 C
13 C
14 C VARIABLES
15 C USED: (NONE)
16 C
17 C MODIFIED: (NONE)
18 C
19 C*****
20 C
21 C
22 C CALL SEMVAL
23 C CALL SEMTYP
24 C CALL MODSET
25 C RETURN
26 C
27 C

```

```

--PROCEDURES--(LO-A)
--NAME--TYPE--ARCS--CLASS--
MODSET      0      SUBROUTINE
SEMTYP      0      SUBROUTINE
SEMVAL      0      SUBROUTINE

```

```

--ENTRY POINTS--(LO-A)
--NAME--ADDRESS--ARCS--
SVSSEN      5B      0

```

```

--STATISTICS--
PROGRAM-UNIT LENGTH      258 - 21
CN STORAGE USED          86688 - 24960
COMPILE TIME              6.817 SECONDS

```

```

100
110
120
130
140
150
160
170
180
190
200
210
220
230
240
250
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460
470
480
490
500
510
520
530
540
550
560
570

```

SUBROUTINE TABDIS

***** THIS ROUTINE GENERATES THE TABULAR OUTPUT FOR SENSITIV. RESULTS. *****

***** CALLED BY: SENSU *****

***** VARIABLES *****

***** USED: ANSWER, FINDST, RTYPE, NSVS, OBJECTU, SENS, SYSEMI, HOLD *****

***** MODIFIED: DCM, I, IPIND, II, IJ, INDEX, J, K, MN, NCOUNT, NSET, NUM, STAR, TEST, TSVS, U *****

***** COMMON: SEMI, ATTRAM, SENS, SYSNAM *****

***** COMMON: SEM3, UNOLD, I20, O10 *****

***** COMMON: SEM4, RTYPE *****

***** COMMON: SEM5, FINDST(50), NSET, INDEXST(50), KIDSET(50) *****

***** COMMON: SYS1, NSVS *****

SAVE

CHARACTER*10 OBJECTU(16), ANSWER279, TSVS(4), SYSLBL, SENSIS, SYSNAM,

* ATTRAM, DUR(0:4)25, TESTIS, RTYPE25

REAL NUM(0:4)

INTEGER STAR

WRITE(3, '(5(1),1X)')

NCOUNT=1

IF(SENS.NE.'SYSTM'.NSET=1

IF(SENS.EQ.'SYSTM')IF(IND=1)INDST(NCOUNT)

CALL NCARD(1, OBJECTU, ANSWER)

PRINT*, ANSWER

IF(SENS.NE.'SYSTM')THEN

CALL NCARD(2, OBJECTU, ANSWER)

PRINT*, ANSWER

II=1

DO 10 I=1,4

WRITE(3, '(1X,4A10)')(OBJECTU(K),K=II,II+3)

II=II+4

CONTINUE

ELSE

PRINT*, 'FOR SYSTEM', SYSNAM

ENDIF

IF(SENS.EQ.'SYSTM')THEN

IF(NSET

ELSE

PA=NSVS

ENDIF

INDEX=NF/4

```

58 DO 20 I=1,INDEX
59 IJ=1
60 DO 10 J=124-3,124
61 IF (SENS.NE.'SYSTEM') THEN
62 TSYS(IJ)=SYSBL(IJ)
63 ELSE
64 TSYS(IJ)='MODE'
65 ENDIF
66 IJ=IJ+1
67 CONTINUE
68 IF (SENS.NE.'SYSTEM') THEN
69 TEST=SENS
70 ELSE
71 TEST=RTYPE
72 ENDIF
73 WRITE(8, '(IX,FS,T15,4(5X,A10)) )TEST,(TSYS(K),K=1,4)
74 DO 30 J=8,20
75 U=0
76 STAR=0
77 DO 40 K=124-3,124
78 IF (U-HOLD(J,K).GT.U) THEN
79 STAR=K
80 U=U-HOLD(J,K)
81 ENDIF
82 CONTINUE
83 NUM(0)=U-HOLD(1,0)
84 DO 50 K=124-3,124
85 IF (K.EQ.STAR) THEN
86 NUM(IJ)=U-HOLD(J,K)
87 DUR(IJ)=1
88 IJ=IJ+1
89 ELSE
90 NUM(IJ)=U-HOLD(J,K)
91 DUR(IJ)=1
92 IJ=IJ+1
93 ENDIF
94 CONTINUE
95 WRITE(8, '(IX,5(5E-4,AS)) )NUM(0),DUR(0),NUM(1),DUR(1),NUM(2),
96 1 DUR(2),NUM(3),DUR(3),NUM(4),DUR(4)
97 CONTINUE
98 IF (INDEX.EQ.1) THEN
99 IJ=1
100 DO 21 I=INDEX+1,INT
101 IF (SENS.NE.'SYSTEM') THEN
102 TSYS(IJ)=SYSBL(IJ)
103 ELSE
104 TSYS(IJ)='MODE'
105 ENDIF
106 IJ=IJ+1
107 CONTINUE
108 IF (SENS.NE.'SYSTEM') THEN
109 TEST=SENS
110 ELSE
111 TEST=RTYPE
112
113
114

```

74174 68700

```

115 ENDIF
116 WRITE(8,'(IX,AE,T15.4(5X,A10))' TEST,(TSVS(K),K=1,IJ=1)
117 DO 31 J=0,20
118   U=0
119   STAR=0
120   DO 41 K=INDEX+1,MN
121     IF(UMOLD(J,K).GT.U) THEN
122       STAR=K
123       U=UMOLD(J,K)
124     ENDIF
125   CONTINUE
126   DUM(8)=UMOLD(J,0)
127   IJ=1
128   DO 51 K=INDEX+1,MN
129     IF(K.EQ.STAR) THEN
130       NUM(IJ)=UMOLD(J,K)
131       DUM(IJ)=.F
132     ELSE
133       IJ=IJ+1
134     ELSE
135       NUM(IJ)=UMOLD(J,K)
136       DUM(IJ)=.
137       IJ=IJ+1
138     ENDIF
139   CONTINUE
140   WRITE(8,'(IX,S(F10.4,AE))' (NUM(I),I=0,IJ-1)
141   CONTINUE
142 ENDIF
143 RETURN
144 END

```

[illegible]

SUBROUTINE TABD16

--PROCEDURES--(LO-A)
 --NAME--TYPE--ARGS--CLASS--

HEADER CHAR10 3 SUBROUTINE
 SVSLBL 1 FUNCTION

--STATEMENT LABELS--(LO-A)
 --LABEL-ADDRESS--PROPERTIES--DEF

10 INACTIVE DO-TERM 48
 19 INACTIVE DO-TERM 67
 20 INACTIVE DO-TERM 100
 21 INACTIVE DO-TERM 110

--LABEL-ADDRESS--PROPERTIES--DEF
 30 INACTIVE DO-TERM 99
 31 INACTIVE DO-TERM 141
 40 INACTIVE DO-TERM 182

--LABEL-ADDRESS--PROPERTIES--DEF
 41 INACTIVE DO-TERM 125
 50 INACTIVE DO-TERM 136
 51 INACTIVE DO-TERM 139

--ENTRY POINTS--(LO-A)
 --NAME--ADDRESS--ARGS--

TABD16 50 0

--STATISTICS--

PROGRAM-UNIT LENGTH 10048 * 516
 CH LABELLED COMMON LENGTH 35618 * 1965
 CH STORAGE USED 626008 * 25384
 COMPILE TIME 0.225 SECONDS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GOR/AA/81D-1	2. GOVT ACCESSION NO. A111 105	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MADAM: MULTIPLE-ATTRIBUTE DECISION ANALYSIS MODEL VOLUME II		5. TYPE OF REPORT & PERIOD COVERED MS Thesis
7. AUTHOR(s) Wayne A. Stimpson, 2Lt, USAF		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT/EN) Wright-Patterson AFB, OHIO 45433		8. CONTRACT OR GRANT NUMBER(s)
11. CONTROLLING OFFICE NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
13. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1981
		13. NUMBER OF PAGES 164
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 28 JAN 1982		
18. SUPPLEMENTARY NOTES APPROVED FOR PUBLIC RELEASE AFR 190-17. FREDRIC C. LYNCH, Major, USAF Director of Public Affairs		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Multiple-Attribute Decision Analysis Computer Aid Preference Value Air Force Institute of Technology (AFIT) Wright-Patterson AFB, OH 45433		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The complex multifaceted decision situations present today suggest the need for a timely, automated tool. A decision-maker is forced into comparing alternative actions or systems over an entire set of different measures of merit. This effort is an on-line, real-time, computer-based decision aid designed to assist the decision-maker in clarifying preferences in a complex decision environment. It is applicable to problems which may be represented by a hierarchy of objectives to be satisfied. The program is MADAM: Multiple-Attribute Decision Analysis Model, and it is written in FORTRAN V and is implemented on the CYBER 175		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

system. MADAM is designed to aid the decision-maker as he or she progresses through problem formulation, parameterization, sensitivity analyses, and a decision, including storage of all data and rationales. Deterministic problems are analyzed through Multi-Attribute Utility Theory concepts and an additive value function is utilized for sensitivity analysis. Pairwise preferential independence is tested between attributes. The sensitivity analyses include a cumulative weight analysis, a relative weight analysis, and an attribute level analysis. The analyses may be conducted by fixing an objective to be considered and conducting the analysis across the alternative systems or actions, or conversely by fixing the alternative to be considered and conducting the analysis across a desired set of objectives.

The work is divided into two volumes. Volume I is a theoretical presentation and includes a user's manual. It requires no programming expertise and may be used independently of Volume II. Volume II is a programming manual including the source code. It may not be used independently of Volume I.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)